

estimate

estimate • analyze • plan • control

Affordability Analysis: The Role of Process, Cost and ROI Modeling In Improved Program Management and Performance

Galorath Incorporated

Daniel D. Galorath: Founder & CEO

galorath@galorath.com



Key Points

Viable
affordability
decisions yield
project
achievements

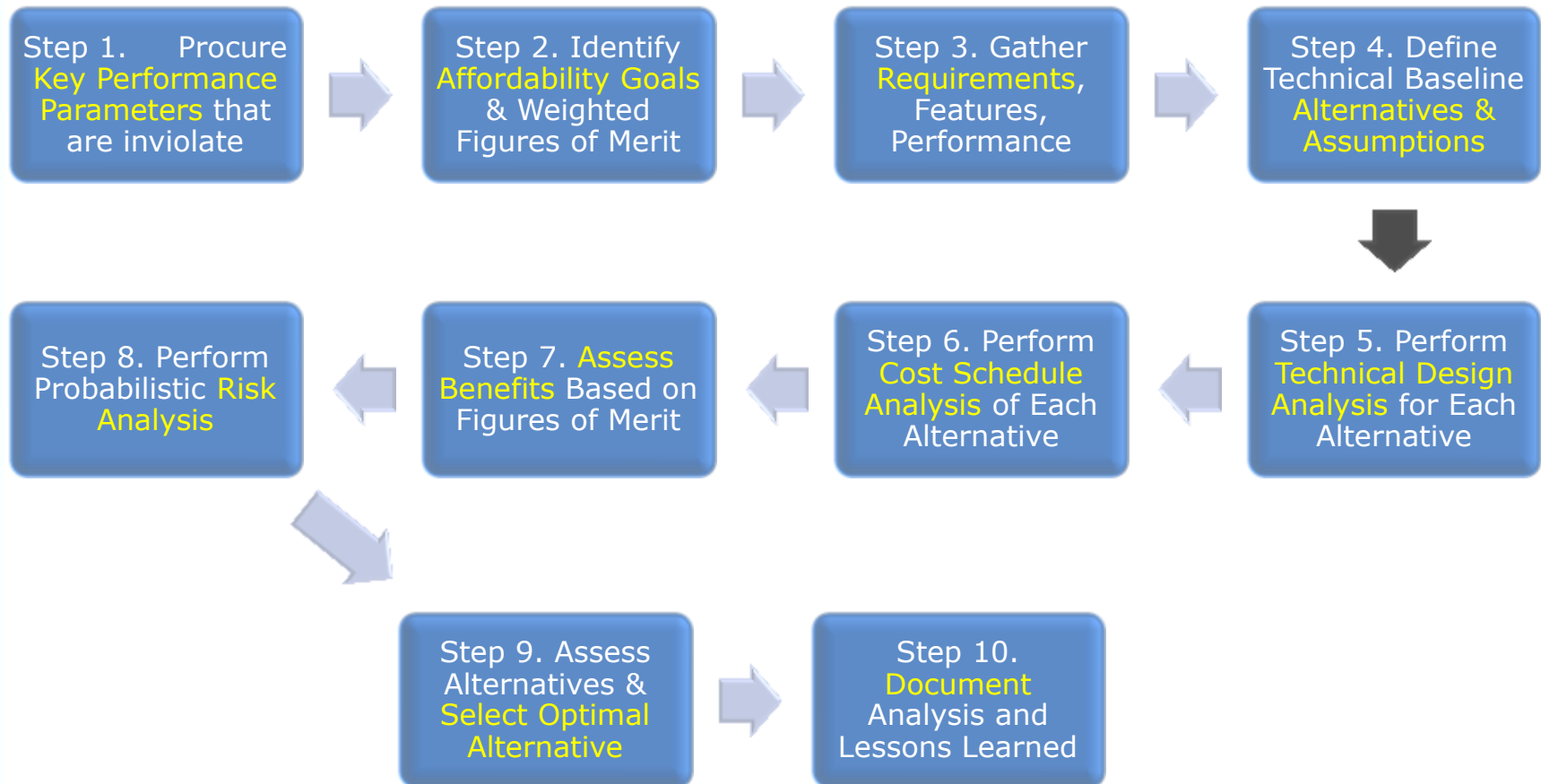


Repeatable
affordability
process is a
key method
of analyzing
affordability

We can make
best value
decisions,
driving down
cost & increasing
value



Galorath Affordability Process 1.3: Use An Affordability Process To Determine Best Value



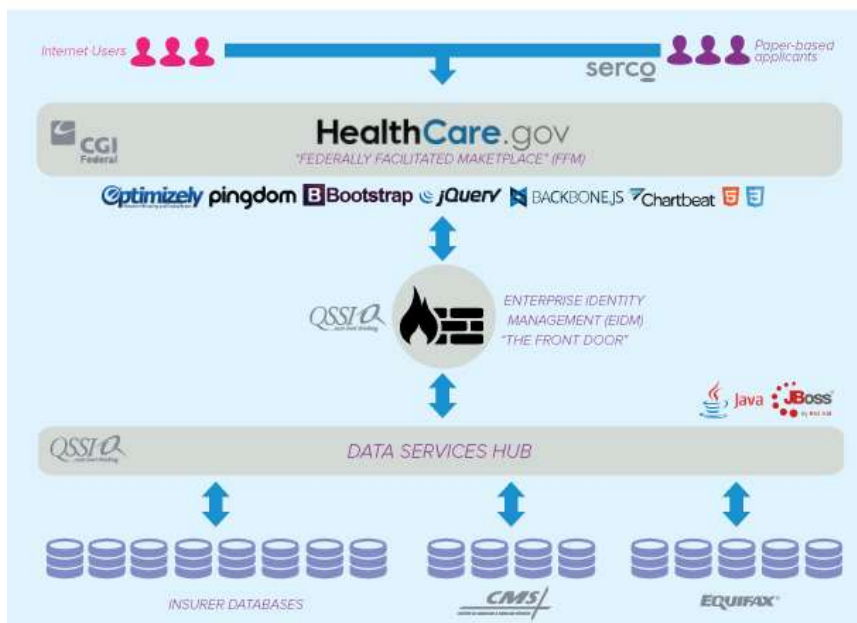
Pricing strategies assumed in step 7. Since price is a figure of merit

HEALTHCARE.GOV

breakdown & technology stack



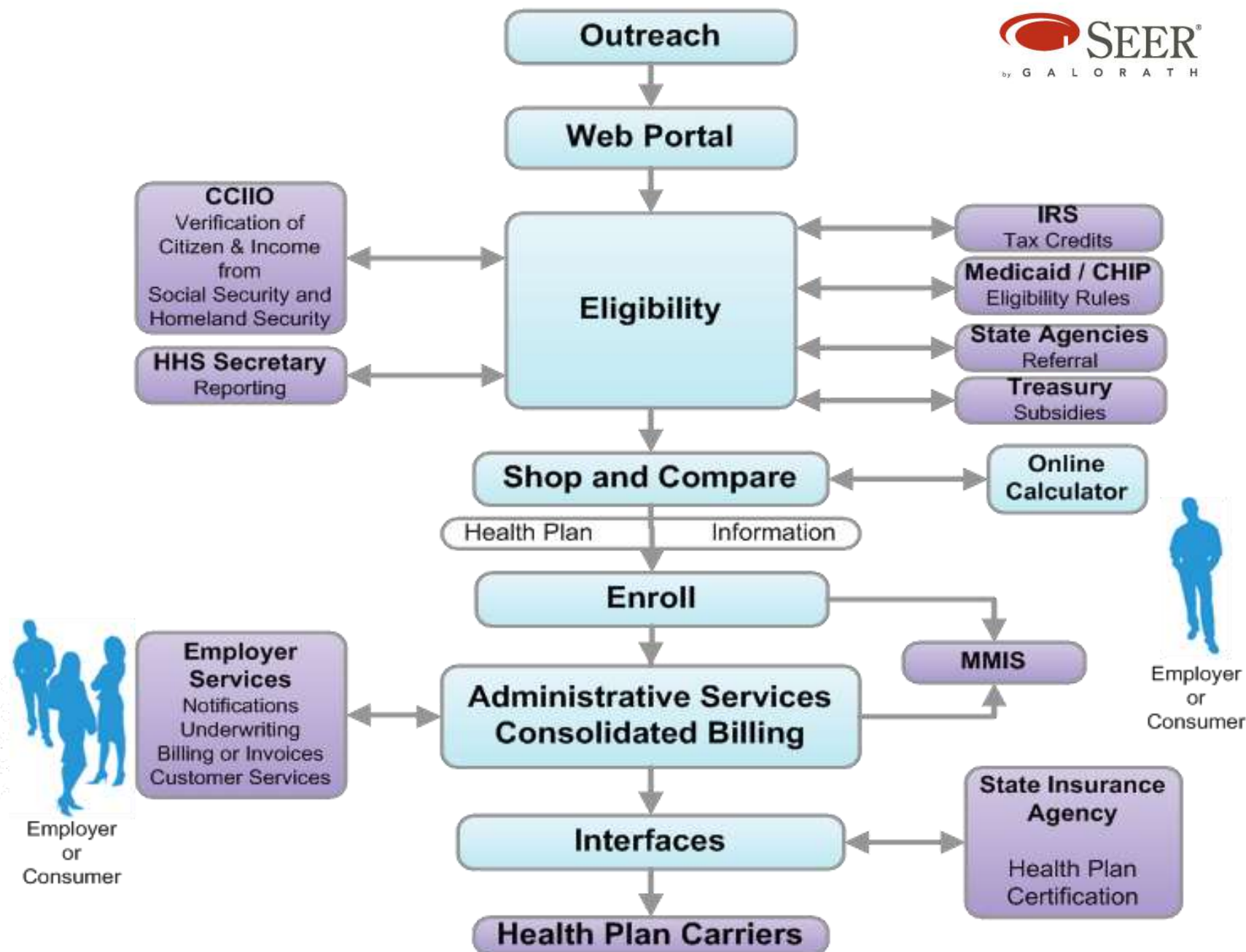
Systems Integrator, End-End Testing, Project Management



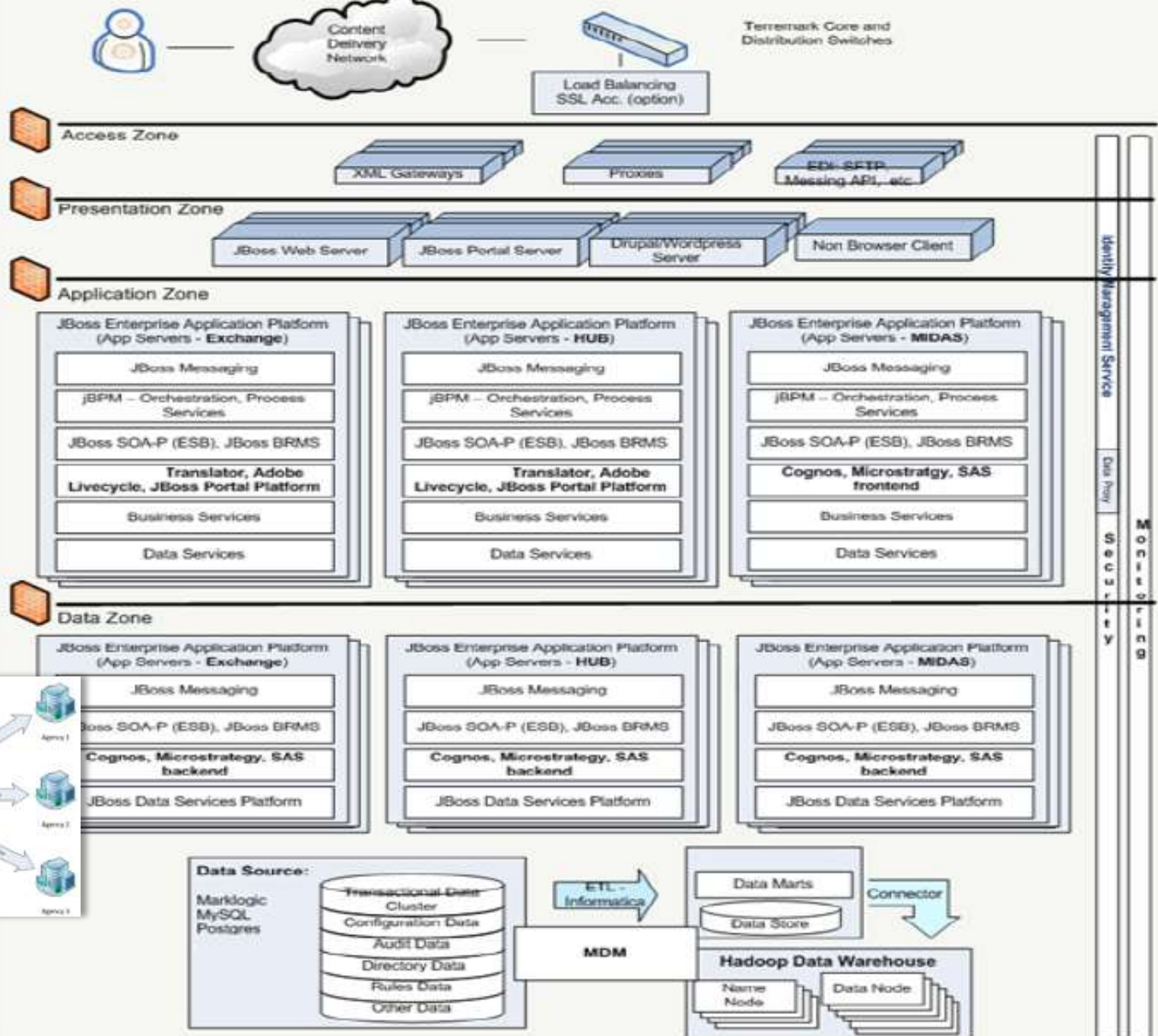
Source:
U.S. House, Committee on Energy and Commerce. PPACA Implementation
Failures: Didn't Know or Didn't Disclose?, Hearing, October 24, 2013.

This work also represents my personal independent research
utilizing standards-based web browser debugging tools.
I do not have any political party affiliations and/or inclinations.

Copyright © 2013 Eduardo Garcia



CMS Enterprise Cloud – Technical Architecture View



3

How much has been spent on Healthcare.gov?

CORRECTION

WAS



Table 4: CMS Obligations for Contracts That Support Federally Facilitated Exchanges (FFE) and Data Hub Establishment by Largest Contractor, through March 31, 2013

Contractor	Examples of activities	Amount obligated (dollars)
CGI Federal Inc	FFE information technology and healthcare.gov	\$87,997,938
Quality Software Services Inc	Data hub	55,098,237
Booz Allen Hamilton	Enrollment and eligibility planning and state grant technical assistance	37,737,550
National Government Services Inc	Consumer call center and Small Business Health Options Program (SHOP) premium aggregations	31,580,846
The Mitre Corporation	Project management and information technology security	22,028,672
Logistics Management Institute	Health plan management, rate analysis, and benefit package review	19,107,667
DEDE Inc DBA Genova Technology	Information technology	16,026,915
Terremark Federal Group	Cloud computing services	15,539,713
IDL Solutions	Enterprise data and design support	9,342,512
Navigant Consulting Inc	Outreach and collection activities	8,949,560

NOW



\$300M+

The Washington Post PostTV Politics Opinions Local Sports National World Busi



The Fact Checker

The Truth Behind The Rhetoric | By Glenn Kessler

Congress Issues Barack Obama Political Ads 2016 Candidates About Archives Follow:   

The Fact Checker

How much did HealthCare.gov cost? (Part 2)

4 How much does \$150M buy?

Assuming....

10-30 major systems



\$18K average monthly contractor cost (government rates)



About 1.3
MILLION
HOURS



2.7m HOURS

Taking
just over
2 YEARS



3 YEARS

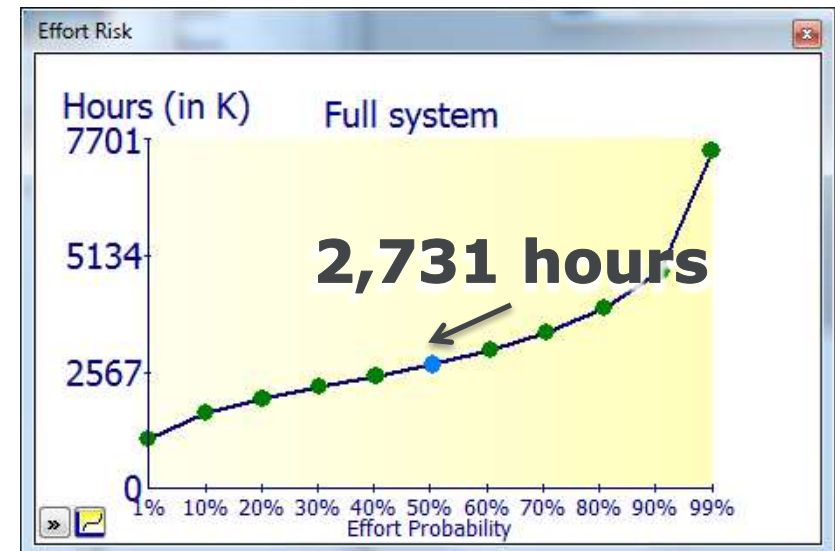
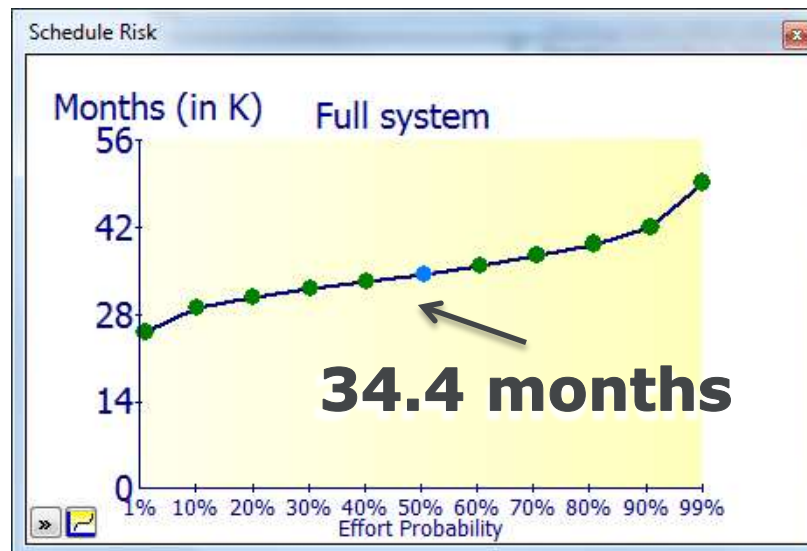
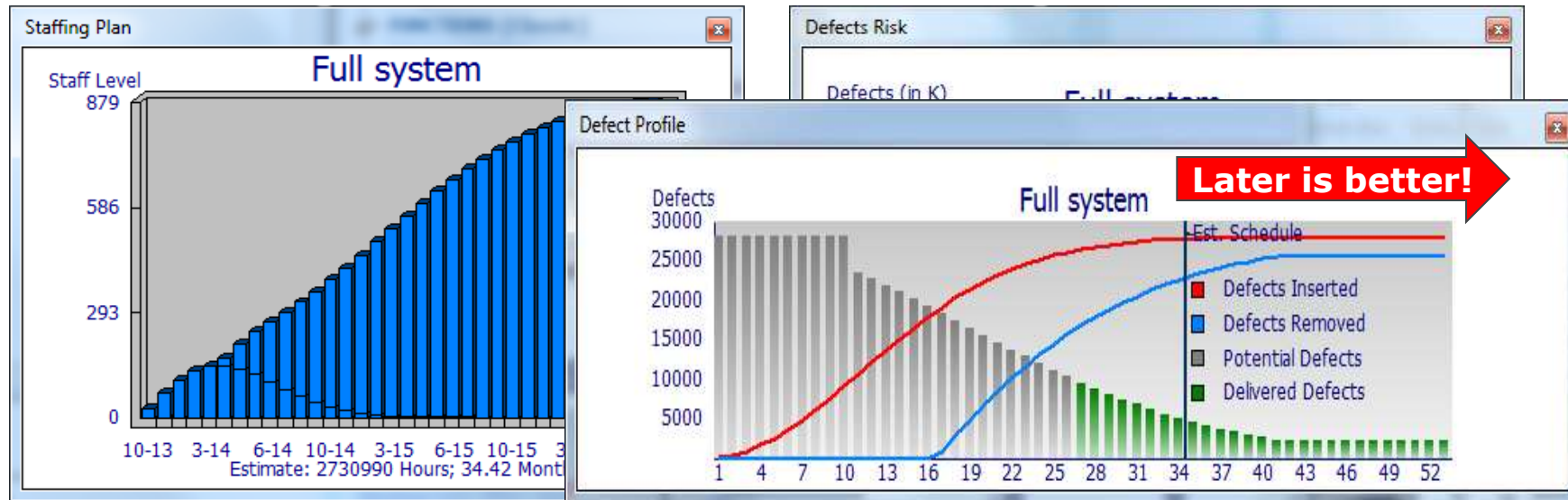
Source: Galorath's SEER-SEM Estimation Model

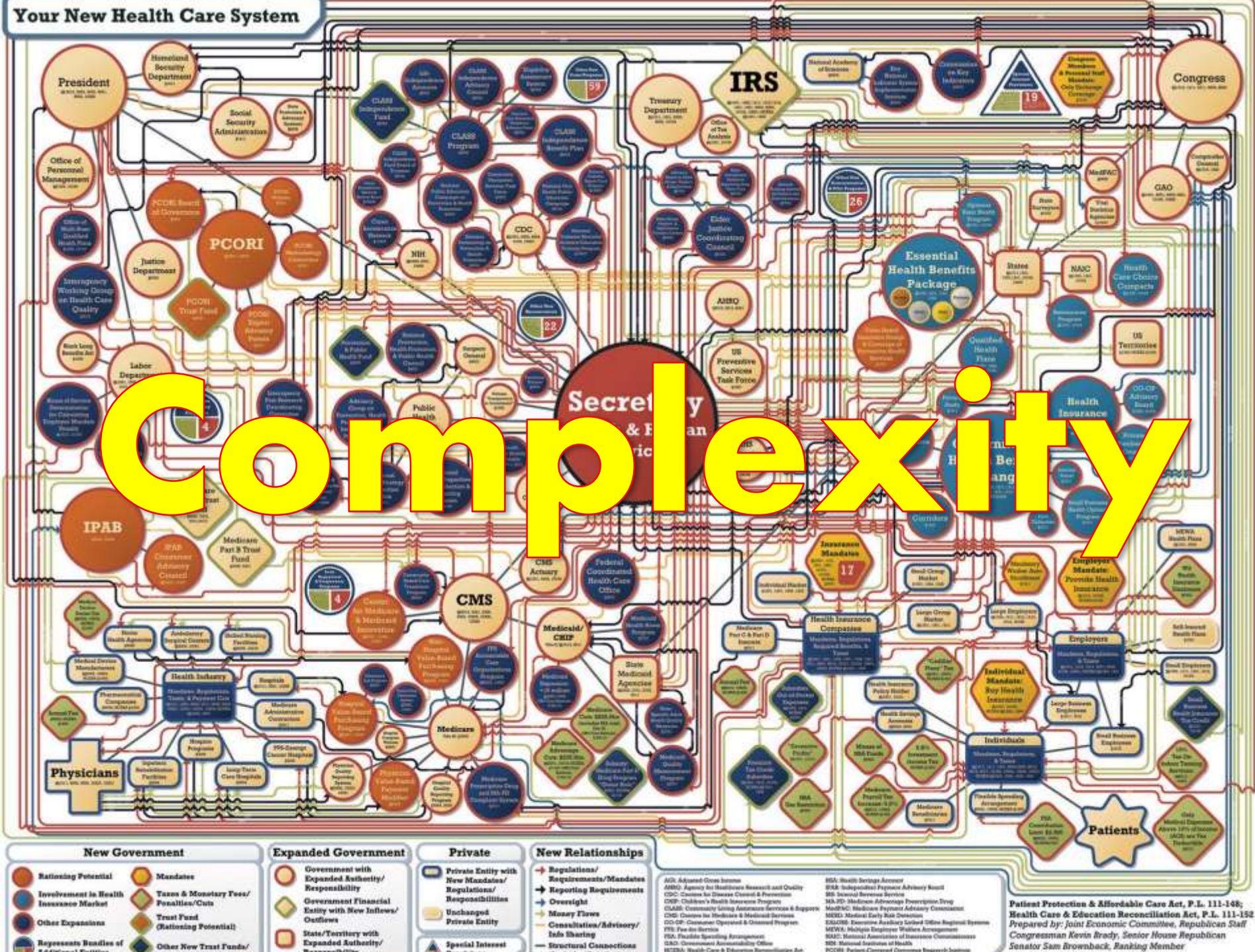


5 What kind of quality is delivered on delivery day?



\$300M Buys About 1M Lines of Code (or 40K Function Points)





Part 1 Health Insurance Marketplace

Part 2
October 1,
2013

Part 3
January 1,
2014

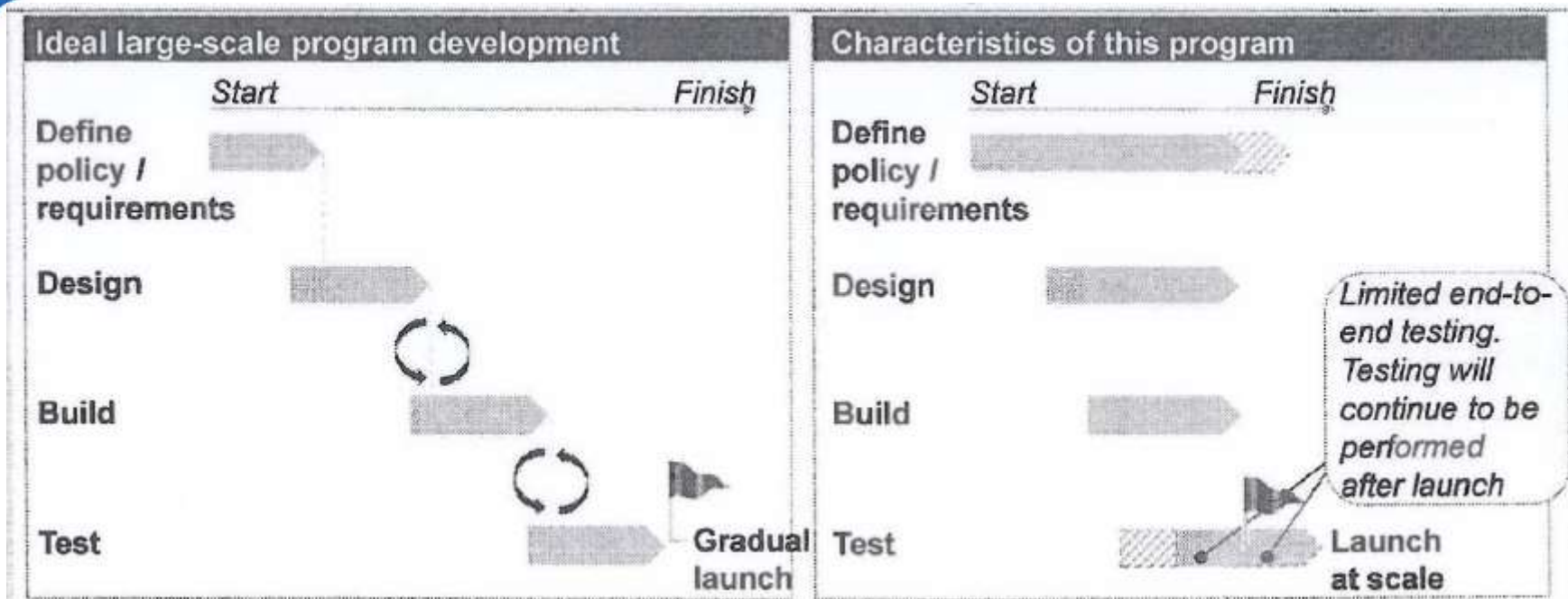
Part 1

Health Insurance Marketplace

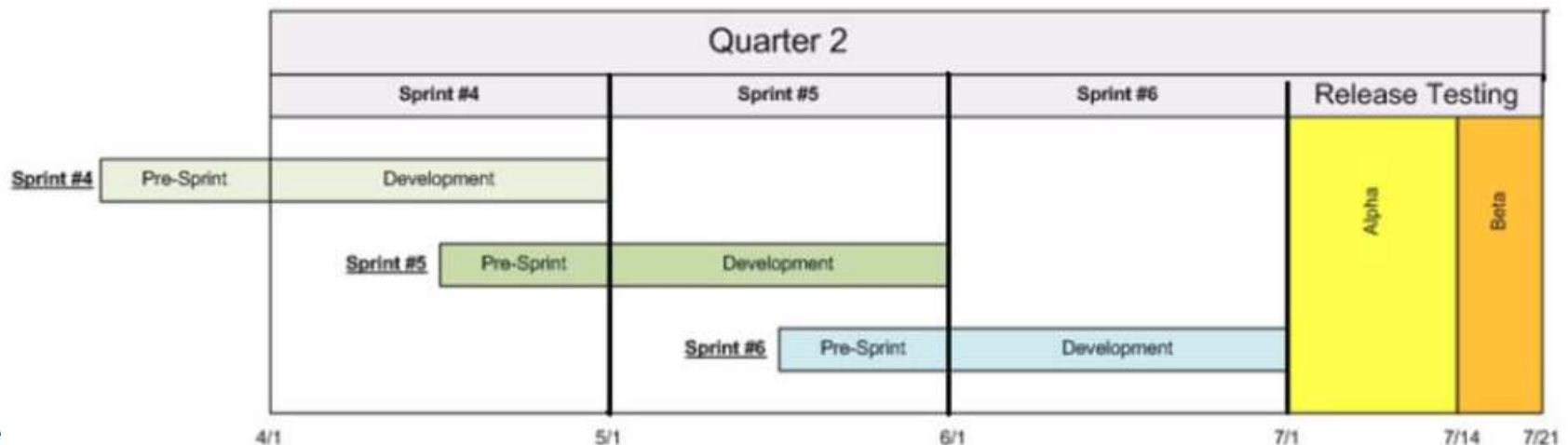
There are 3 key dates you'll want to mark on your calendar:

- **October 1, 2013.** Marketplace open enrollment started

UAT	CCIO	██████████	Open E&E UAT - Testing involving analysis of the information dependent on the 22 logic drivers of eligibility and enrollment. Testers will use QSSI Test Scripts delivered to walk through a logical progression of as many screens as possible in an effort to validate the functionality delivered.	10/01/2013 - 10/04/2013
UAT	OC	██████████	UAT for E&E Consumer Web Experience and Functionality - JOLT 11	
UAT	SHOP	██████████	No report (testing not started)	



Process Does Not Guarantee Success



Step 1 Key Performance Parameters (KPPs)

Step 1. Procure
Key Performance
Parameters that
are inviolate



- **Key Performance Parameters Defined:** Critical subset of performance parameters, capabilities and characteristics **so significant that failure to meet them can cause concept or system selected to be reevaluated or the project reassessed or terminated.** (Adapted from Glossary of Defense Acquisition)

KPP Example Criteria

Essential for
defining the
required
capabilities?

Contributes to
significant
improvement in
the operational
capabilities of the
enterprise?

Achievable and
affordable?

Measurable and
testable/verifiable?

Can KPP attribute
be analyzed
throughout the life
cycle?

If not met, will the
sponsor of the
project be willing
to cancel or
significantly
restructure the
project?

Should These Have Been KPP's (Cloud Black Swan Examples)

<http://www.datacenterknowledge.com/archives/2012/12/05/the-cloudy-side-of-cloud-computing/>

- **Security & Breaches:** Anticipate growing Malicious attacks and accidental data loss
- **Outages:** 2007- late 2012 **568 hours downtime** between 13 major cloud carriers. Cost the customer base about **\$72 million** (International working group on cloud computing resiliency)
- **Learning curve:** Successful cloud model takes knowledge around multiple technological disciplines. Once in place, however, managing can also be issue
- **Vendor lock-in:** Migrating cloud environment to another provider difficult... Not often considered
- **Data portability and porting costs**
- **Software modification Costs (PaaS)**
- **Software Setup (SaaS)**

Step 2. Identify Weighted Affordability Goals & Figures of Merit

- **Figure of merit:** A quantity used to characterize the performance of a device, system or method, relative to its alternatives e.g.
 - Cost
 - Response time of a computing action
 - Survivability
 - Calories in a serving
 - Resolution of a digital camera
 - Battery life
 - Coverage

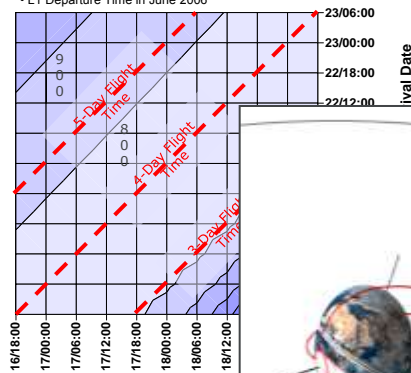
Used to compare alternatives
For example more cheaper UAVs may provide
better coverage for the same \$ than fewer more
powerful UAVs

Key Figures of Merit (Source NASA Space Systems Engineering)

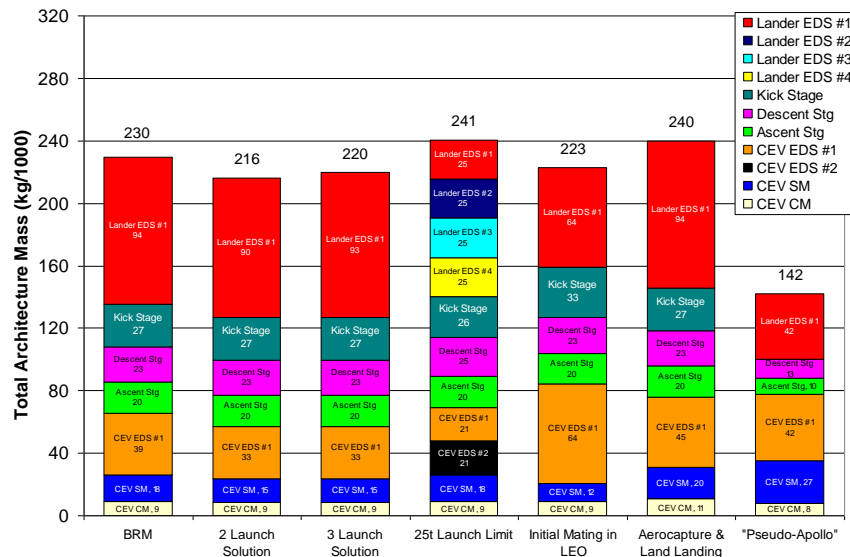
Mission Design

L1-Earth Co-Planar Inbound Delta V Requirement (m/s)

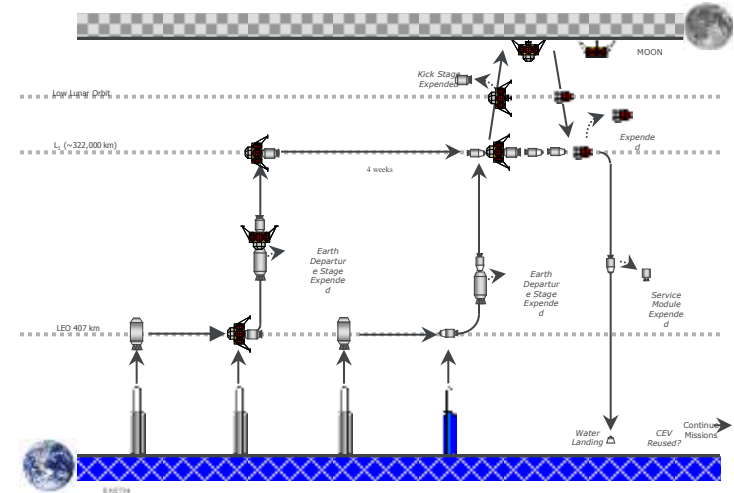
- Moon: Inclination near maximum, Distance near perigee
- L1 Departure Time in June 2006



Initial Mass in LEO



Reference Operations Concept



Key Figures of Merit

Safety

- # of Critical Events
- Mission Complexity
 - Abort Options
 - Crew Time
- Technology Risk
- Probability of launch success
 - Etc.

Effectiveness

- Total Mass
- Dry Mass
- Surface Time
 - Etc.

Extensibility

- Long-Stays
 - Mars
- Other destinations
 - Etc.

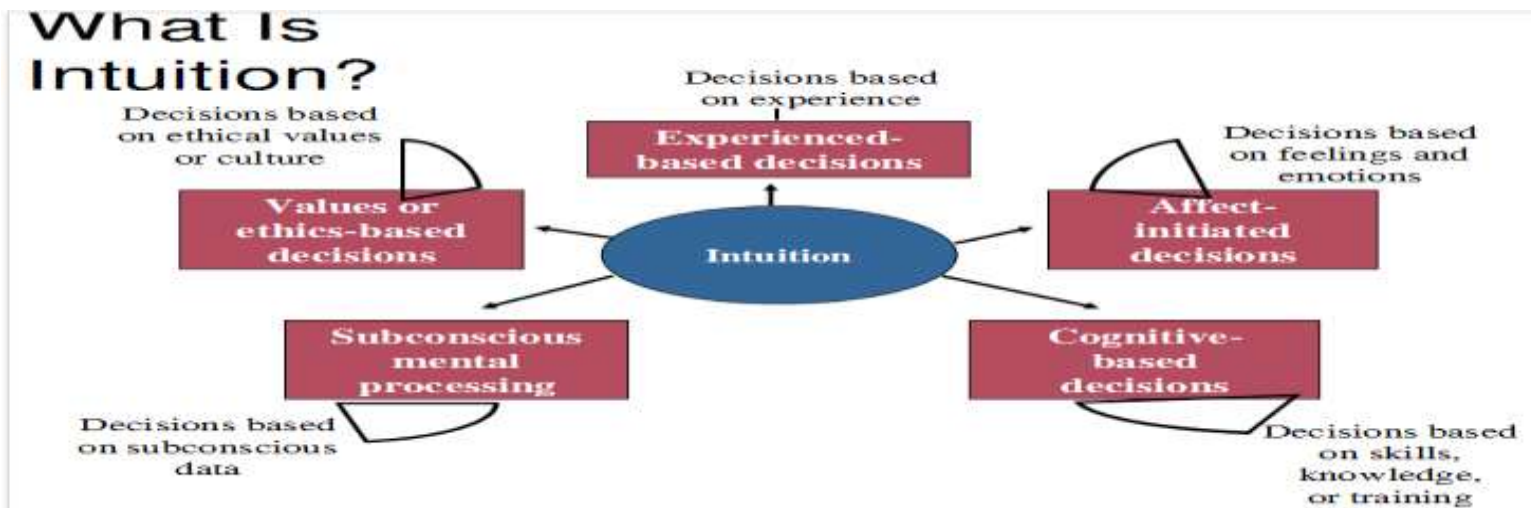
Cloud Example: But When We Look at Figures of Merit

- Is the cloud secure enough?
- Is the cloud fast enough?
- Is cloud vender reliable enough?
- Other figures of merit for this system?

Every case is different
We can't say cloud or on-premises is always better

Building Weightings

- Allocate weights to each figure of merit IN advance
 - KPPs should be ok'ed to get here
- Gives appropriate priority to each
- Consider using expected value when decisions are financial
- Intuition can be valuable but is not repeatable



What is intuition: Source Unknown

Step 3 Gather Requirements, Features, Performance

Step 3. Gather
Requirements,
Features,
Performance



- Functional requirements:
Describe interactions between the system environment independent of implementation
 - Watch system must display time based on location
- Nonfunctional requirements: User visible aspects of the system not directly related to functional behavior
 - Response time must be less than 1 second
 - Accuracy must be within a second
 - Watch must be available 24 hours a day except from 2:00am-2:01am and 3:00am-3:01am
- Groundrules: Imposed by the client or the environment in which the system will operate
 - The implementation language must be COBOL.
 - Must interface to the dispatcher system written in 1956

Data-Gathering Techniques¹

Technique	Good for	Kind of data	Plus	Minus
Questionnaires	Answering specific questions	Quantitative and qualitative data	Can reach many people with low resource	The design is crucial. Response rate may be low. Responses may not be what you want
Interviews	Exploring issues	Some quantitative but mostly qualitative data	Interviewer can guide interviewee. Encourages contact between developers and users	Time consuming. Artificial environment may intimidate interviewee
Focus groups and workshops	Collecting multiple viewpoints	Some quantitative but mostly qualitative data	Highlights areas of consensus and conflict. Encourages contact between developers and users	Possibility of dominant characters
Naturalistic observation	Understanding context of user activity	Qualitative	Observing actual work gives insight that other techniques cannot give	Very time consuming. Huge amounts of data
Studying documentation	Learning about procedures, regulations, and standards	Quantitative	No time commitment from users required	Day-to-day work will differ from documented procedures

[1] Preece, Rogers, and Sharp "Interaction Design: Beyond human-computer interaction", p214

Step 4. Define Technical Baseline Alternatives & Assumptions

Step 4. Define
Technical Baseline
Alternatives &
Assumptions



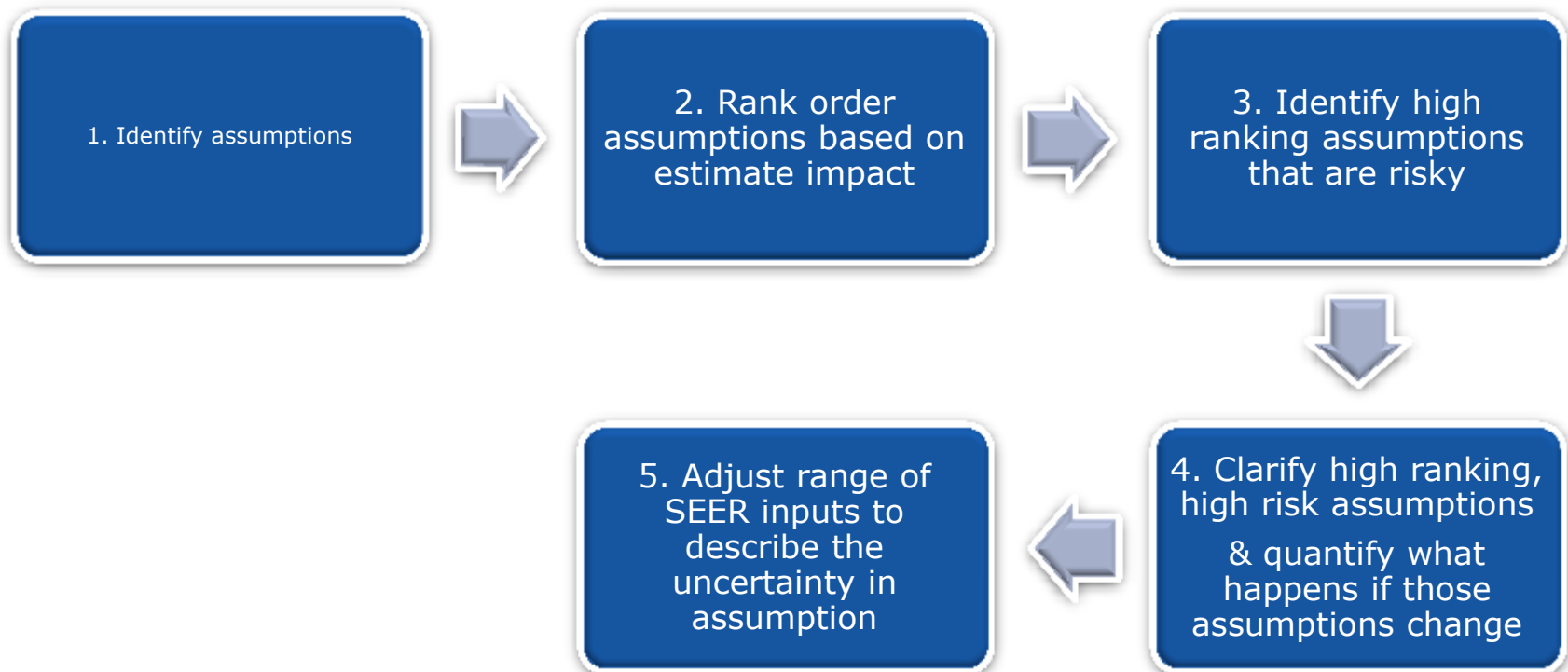
- Functionality included in the estimate or range must be established
 - Defines technical goals, objectives, and scope and provides the basis for estimating project cost and schedule. is managed and communicated in a structured and planned way DAU
 - A living, revised document, set of documents, database, etc.
 - When detailed functionality is not known, groundrules and assumptions state what is and isn't included in the estimate
 - Issues of COTS, reuse, and other assumptions should be documented as well

Ground Rules & Assumptions

- Groundrule: given requirement of the estimate (e.g. software must support windows and Linux)
- Assumption: assumed to scope estimate
- Groundrules and assumptions form the foundation of the estimate
 - Early they are preliminary & rife with uncertainty
 - they must be credible and documented
 - Review and redefine these assumptions regularly as the estimate moves forward
- What's known, what's unknown
- Anything relating to scope
 - What's included, what's excluded
- Anything relating to modeling inputs
 - Who you interviewed and when
 - What you learned

Dealing With the “Problem of Assumptions”

- Assumptions are essential but... Incorrect assumptions can drive an estimate to uselessness
- Use an assumption verification process



Step 5 Perform Technical Design Analysis For Each Alternative

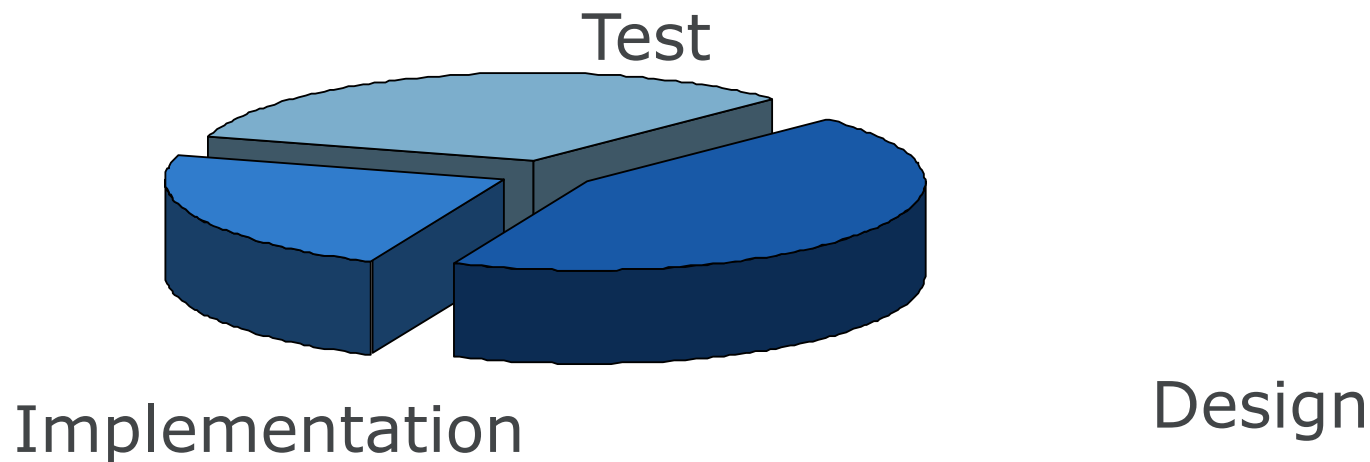
Step 5. Perform
Technical Design
Analysis for Each
Alternative



- Functions needed to satisfy requirements
- For example, to perform any science measurement you will need
 - Sensor (detector system)
 - Power the sensor (power system)
 - Read data from the sensor (data acquisition system)
 - Store data (data archive system)
 - Control sensor, readout, storage (control system)
 - Analyze data (ground data system)
- COTS, Reused, GOTS, New Development, etc.
- These functions will also need to have a set of requirements specified
 - Power system shall supply volts & milliamps to the sensor, data acquisition, archive and control systems

Reuse: Watch Out For Low Cost Assumptions on "Heritage"

- Reuse or Heritage: applying existing software to a new mission (or additional innovation in its current mission)
- Effort to reuse software is routinely under estimated



Why should we care: Bad heritage assumptions often cause major schedule / cost overruns

IT Services Costs Must Consider Service Level Required

- High profile public system will have limited tolerance for down time
- Plan for equivalent of gold SLA when staffing operational support

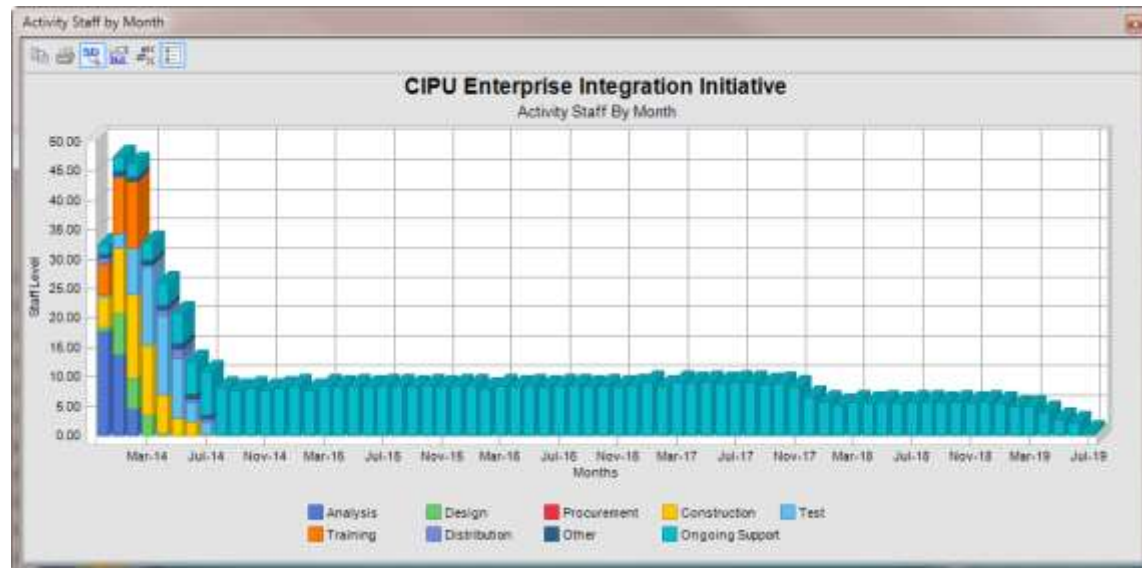
Service Level Agreement (Application Support)

Author: [Name] [Title] [Phone] [Email]

Service Level Target Benchmark for application support, often measured as a percentage of calls answered within a defined timeframe. Note that specific targets and SLA levels vary widely depending on the industry and the nature of the help desk.

Rating	Description
Very High	Acknowledged within 15 minutes and fixed within 2 hours (Gold).
High	Acknowledged within 1 hour and fixed within 4 hours.
Normal	Acknowledged within 4 hours and fixed within 8 hours (Silver).
Low	Acknowledged within 24 hours and fixed within 48 hours (Bronze).
Very Low	Acknowledged within 2 business days and fixed within 5 business days (Basic).

Each rating may be modified by a plus or minus to indicate actual ratings that are slightly higher or lower than what is indicated on the scale. For example, Normal+ would be slightly higher than Normal and Normal- would be slightly lower.



Up front testing needs more people.... Support must keep people ready to support users

IT Services Costs Must Consider Service Level Required

- High profile public system will have limited tolerance for down time
- Plan for equivalent of gold SLA when staffing operational support

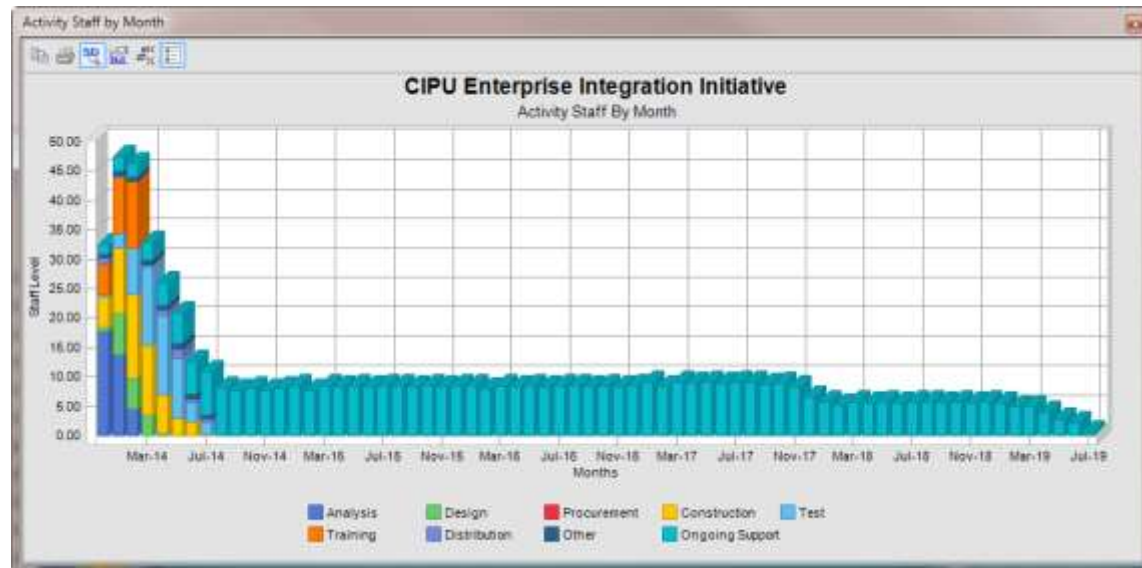
Service Level Agreement (Application Support)

Author: [Name] [Title] [Email]

Service Level Target Benchmark for application support, often measured as a percentage of calls answered within a defined timeframe. Note that specific targets and SLA levels vary widely depending on the industry and the nature of the help desk.

Rating	Description
Very High	Acknowledged within 15 minutes and fixed within 2 hours (Gold).
High	Acknowledged within 1 hour and fixed within 4 hours.
Normal	Acknowledged within 4 hours and fixed within 8 hours (Silver).
Low	Acknowledged within 24 hours and fixed within 48 hours (Bronze).
Very Low	Acknowledged within 2 business days and fixed within 5 business days (Basic).

Each rating may be modified by a plus or minus to indicate actual ratings that are slightly higher or lower than what is indicated on the scale. For example, Normal+ would be slightly higher than Normal and Normal- would be slightly lower.



Up front testing needs more people.... Support must keep people ready to support users

Minimal User Skill Increases Support Required

- Tier 1 support is inversely proportional to user training and skill
- Users will have no prior knowledge of system or procedures which will drive help desk staffing

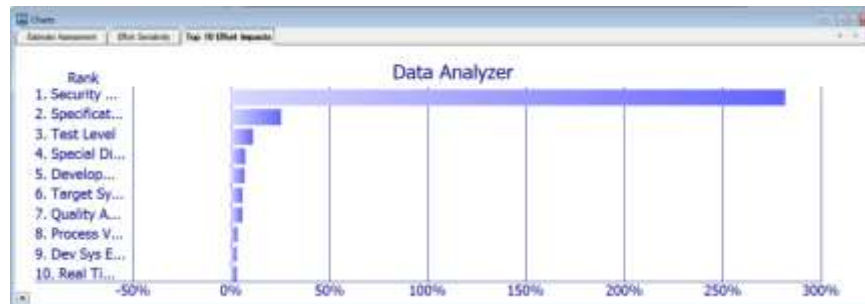
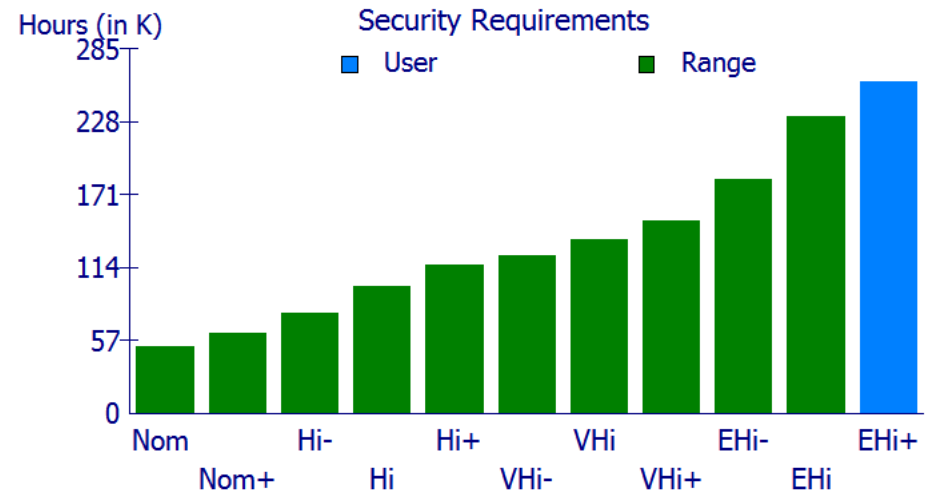


Plan for this pain even
if the system runs perfectly

Software Implemented Security and Safety Requirements Add Significant Cost & Schedule



Rating	Description
Extra Hi+	Class A1: Security formally verified by mathematical proof. (Extremely rare). DO178B - Level A: Software whose anomalous behavior, as shown by the system safety assessment process, would cause or contribute to a failure of system function resulting in a catastrophic failure condition for the aircraft.
Extra Hi-	Common Criteria - EAL 7: Formally Verified Design and Tested. The formal model is supplemented by a formal presentation of the functional specification and high level design showing correspondence. Evidence of developer "white box" testing and complete independent confirmation of developer test results are required. Complexity of the design must be minimized.



Why should we care: Software implemented security and safety requirements can drive costs thru the roof

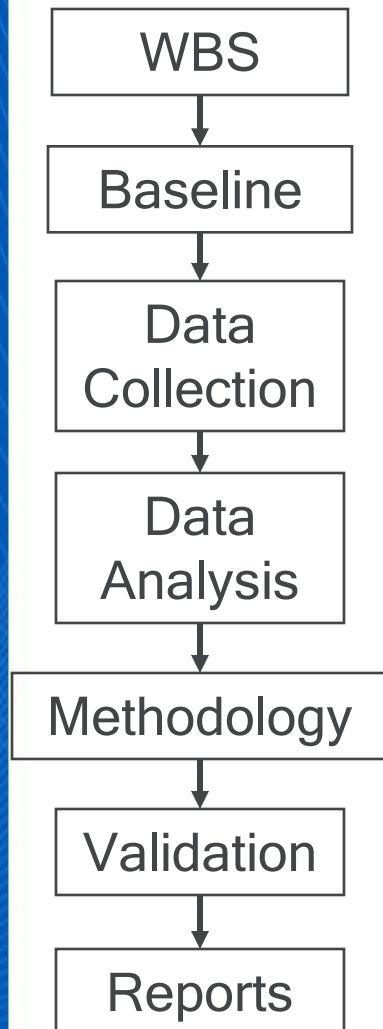
Step 6. Perform Cost Schedule Analysis of Each Alternative

Step 6. Perform Cost Schedule Analysis of Each Alternative

- Estimating is critical for all kinds of systems
 - Yet many treat it as a second rate process
- Everyone estimates.... Just most get it wrong and don't have a process
- Having a repeatable estimation process is critical to both estimating AND to successful projects
- Estimation and measurement go hand in hand

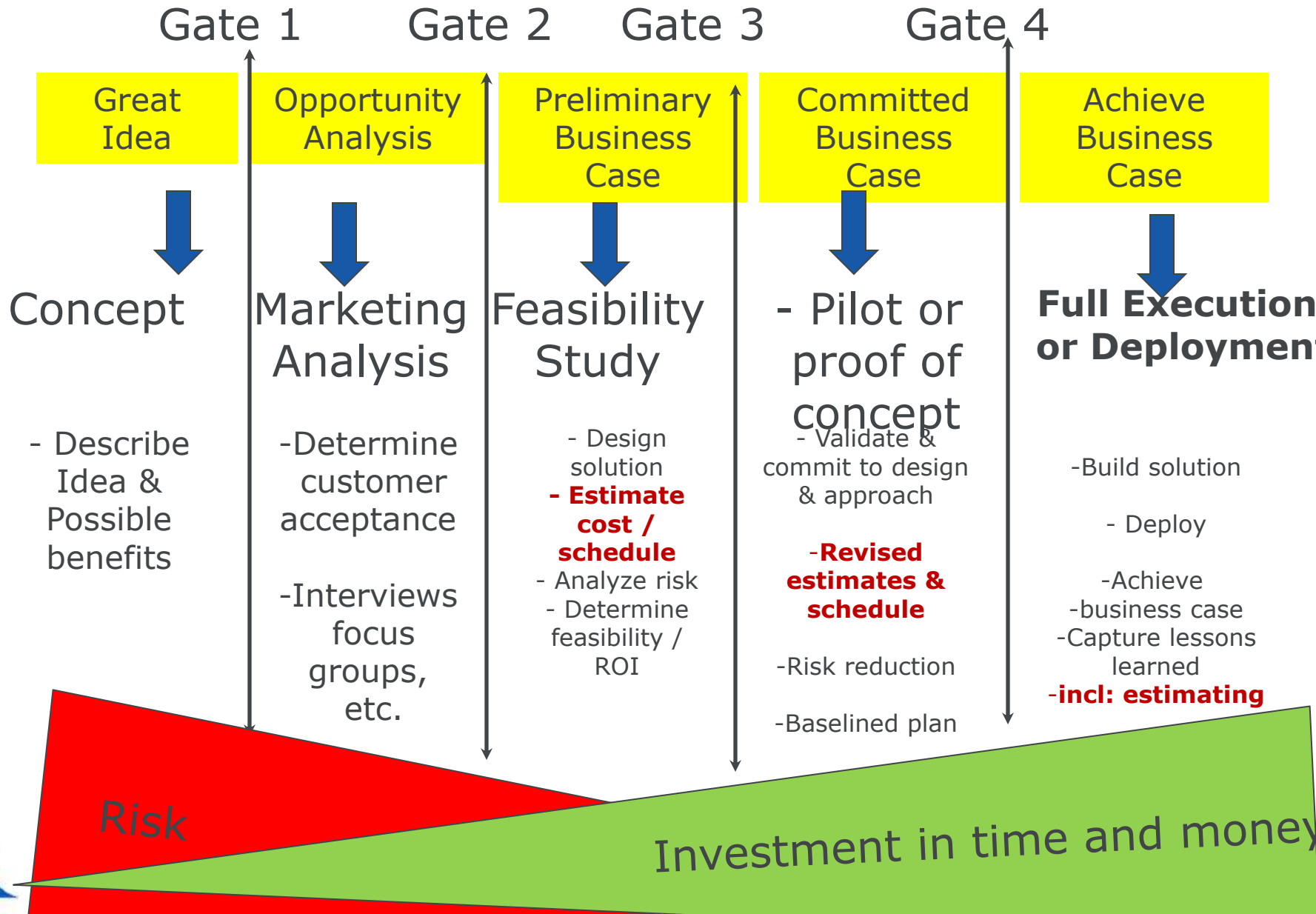
Use An Estimating Process (Generalized 10 Step System Estimation Process 2011)





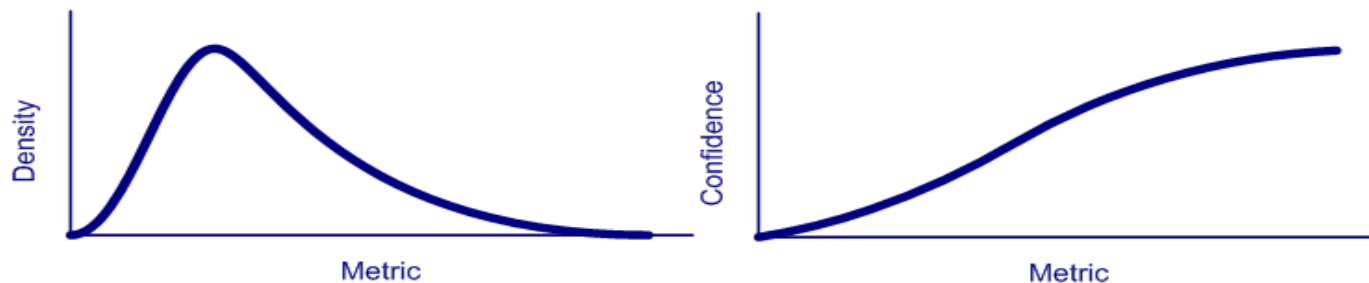
- *Work Breakdown Structure (WBS) Development*
- *Program/System Baseline Development*

Estimating Core Governance Component - A standard Corporate Governance Model (Source: K. Aguanno)



Bad Estimates Are A Root Cause of Project Failure

- An **estimate** is the most knowledgeable statement you can make **at a particular point in time** regarding:
 - Effort / Cost
 - Schedule
 - Staffing
 - Risk
 - Reliability
- Estimates more precise with progress
- ***A WELL FORMED ESTIMATE IS A DISTRIBUTION***



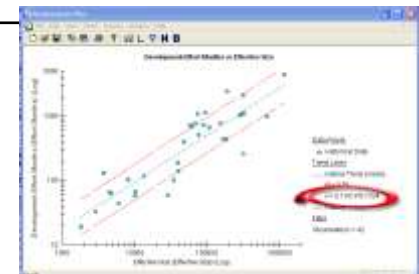
Estimation Methods - 1 of 2



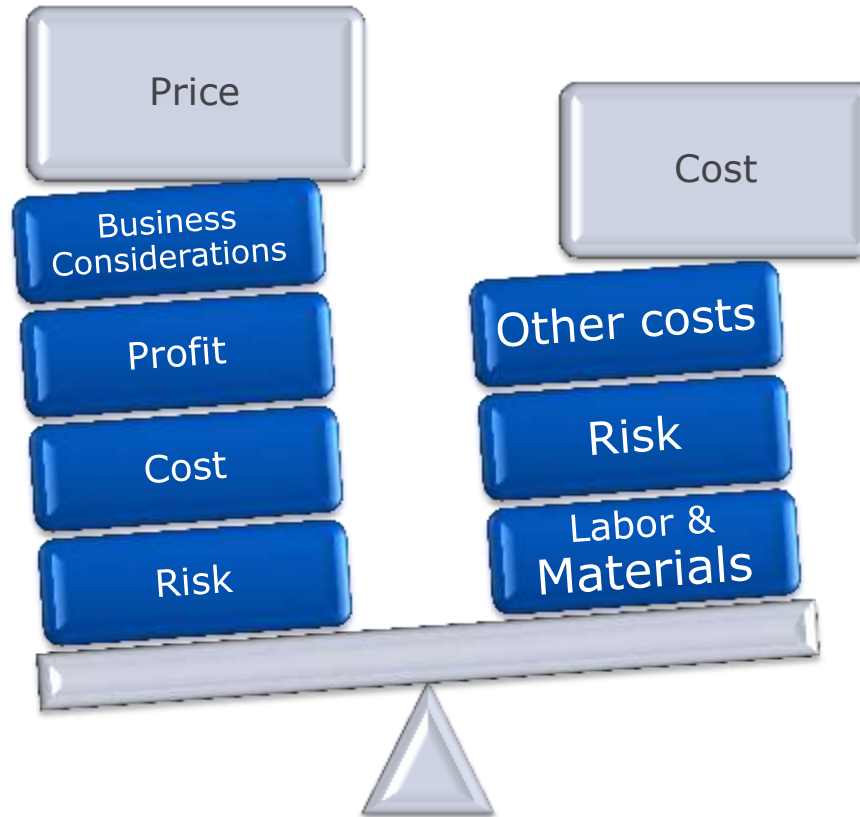
Model Category	Description	Advantages	Limitations
Guessing	Off the cuff estimates	Quick Can obtain any answer desired	No Basis or substantiation No Process Usually Wrong
Analogy	Compare project with past similar projects.	Estimates are based on actual experience.	Truly similar projects must exist
Expert Judgment	Consult with one or more experts.	Little or no historical data is needed; good for new or unique projects.	Experts tend to be biased; knowledge level is sometimes questionable; may not be consistent.
Top Down Estimation	A hierarchical decomposition of the system into progressively smaller components is used to estimate the size of a software component.	Provides an estimate linked to requirements and allows common libraries to size lower level components.	Need valid requirements. Difficult to track architecture; engineering bias may lead to underestimation.

Estimation Methods - 2 of 2

Model Category	Description	Advantages	Limitations
Bottoms Up Estimation	Divide the problem into the lowest items. Estimate each item... sum the parts.	Complete WBS can be verified.	The whole is generally bigger than the sum of the parts. Costs occur in items that are not considered in the WBS.
Design To Cost	Uses expert judgment to determine how much functionality can be provided for given budget.	Easy to get under stakeholder number.	Little or no engineering basis.
Simple CER's	Equation with one or more unknowns that provides cost / schedule estimate.	Some basis in data.	Simple relationships may not tell the whole story. Historical data may not tell the whole story.
Comprehensive Parametric Models	Perform overall estimate using design parameters and mathematical algorithms.	Models are usually fast and easy to use, and useful early in a program; they are also objective and repeatable.	Models can be inaccurate if not properly calibrated and validated; historical data may not be relevant to new programs; optimism in parameters may lead to underestimation.



Remember Cost and Price Are Different (Adapted from Morton)



- **Price:** Amount Charged to Customer (considering cost, profit, risk, Price to win, business considerations, etc.)
 - e.g. New Car - Discounts
 - e.g. Machinists - Idle
 - e.g. Golden Gate Bridge - Cables
 - e.g. NASA – Photos

Affordability Initiatives With “Should Cost” and “Will Cost”



Many View Bottoms up estimates as the requirement
for Should Cost / Will Cost Analysis

But parametrics can do analysis faster as well as
provide more tradeoffs

Example: Project Cost Alone Is not The Cost of IT Failure (Source: HBR)



- Case Study: Levi Strauss
 - \$5M ERP deployment contracted
 - Risks seemed small
 - Difficulty interfacing with customer's systems
 - Had to shut down production
 - Unable to fill orders for 3 weeks
- **\$192.5M charge against earnings on a \$5M IT project failure**



“IT projects touch so many aspects of organization they pose a new singular risk”

Step 7. Assess Benefits Based on Figures of Merit

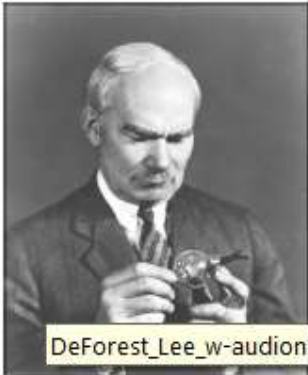
Step 7. **Assess**
Benefits Based on
Figures of Merit

- Return on Investment often main criterion in IT systems

Future cost of defects remaining in code at release, a component of the cost of ownership:

- **Principal** – Cost of fixing problems remaining in the code after release that must be remediated
- **Interest** - Continuing IT costs attributable to the violations causing technical debt, including higher maintenance costs, greater resource usage, etc.
- **Liability**—business costs related to outages, breaches, corrupted data, etc.
- **Opportunity cost**—benefits that could have been achieved had resources been put on new capability rather than retiring technical debt

While Optimism Needs Tempering, So Does Short Sightedness (Source Northrop)



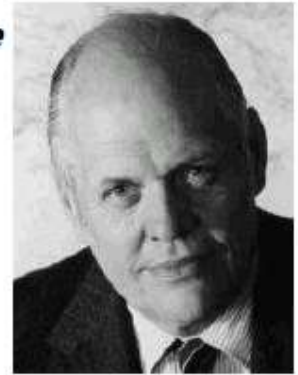
DeForest_Lee_w-audion

"Man will never reach the moon regardless of all future scientific advances."

- Dr. Lee DeForest, Inventor of Television

"There is no reason anyone would want a computer in their home."

- Ken Olson, president and founder of Digital, 1977



"Airplanes are interesting toys but of no military value."

- Marechal Ferdinand Foch, Professor of Strategy, Ecole Superieure de Guerre

"640K ought to be enough for anybody."

- Bill Gates, 1981



"Any general who's worth his salt knows that war is not a Nintendo game, war is not something that's fought by robots."

- Norman Schwarzkopf, 1991

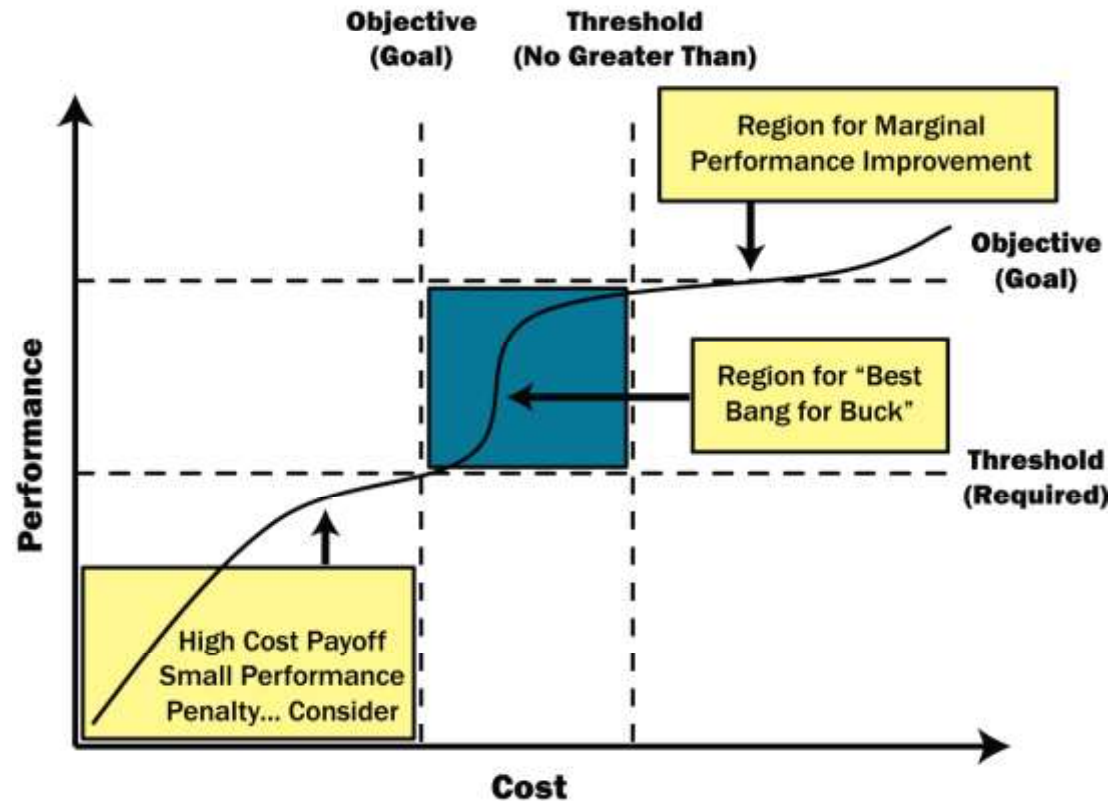
"To throw bombs from an airplane will do as much damage as throwing bags of flour. It will be my pleasure to stand on the bridge of any ship while it is attacked by airplanes."

- Newton Baker, Sec. of War, 1921



Affordability Trades (Source NASA Space Systems Engineering)

“Best Bang for the Buck”



Augustine’s Law of Insatiable Appetites
The last 10 percent of performance generates
 $\frac{1}{3}$ of the cost and $\frac{2}{3}$ of the problems.

Example: Cloud Economics Fall Apart When Application Needs Rewrite for Cloud



- Rewriting applications to make them work in the cloud
- [Dave Linthicum](#), who also participated in Dana's latest analyst roundtable, points out that there's a lot more to enterprise IT than simply accessing and running applications.
- "Cloud computing typically is going to be a better, more strategic, more agile architecture, but it's also typically going to be more expensive, at least on the outcome," Can be lots of costly infrastructure changes Dave Linthicum

Step 8 Perform Risk Analysis

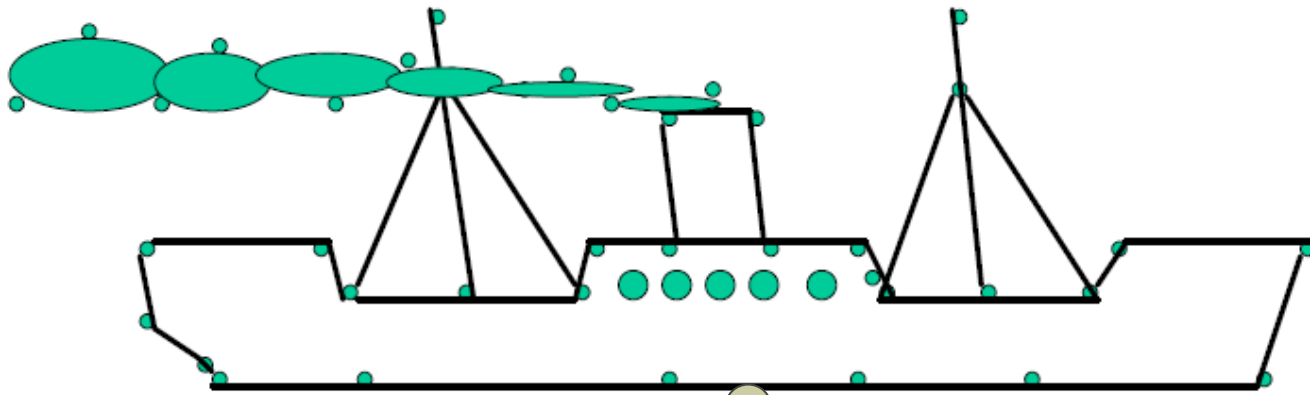
- A viable risk analysis may point out different decisions than simple analysis

Step 8. Perform
Probabilistic Risk
Analysis

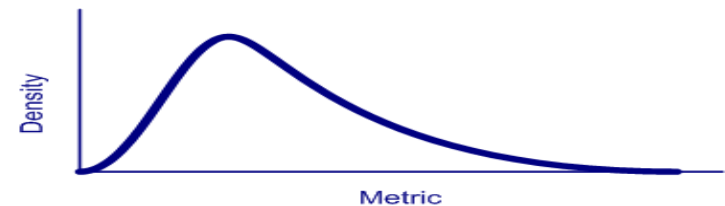
System Description (Parametrics Can Estimate More, Earlier)

Adapted from CEBOK

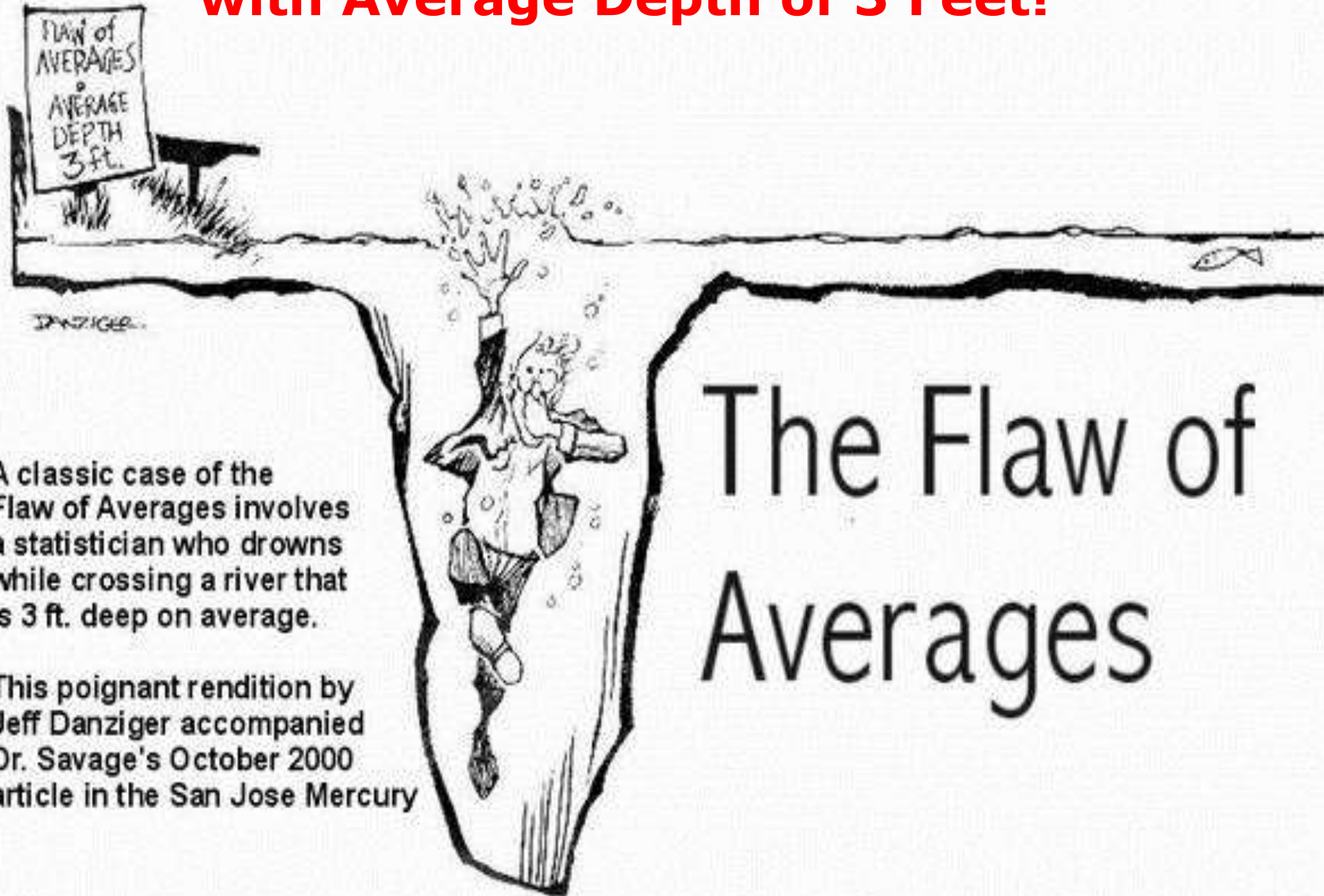
“If you can’t tell me what it is,
I can’t tell you what it costs.”
-Mike Jeffers



“If you can tell me the range of
what it might be, I can tell you the
range of cost, schedule &
probability.”
-Dan Galorath



Statistician Drowns in River with Average Depth of 3 Feet!

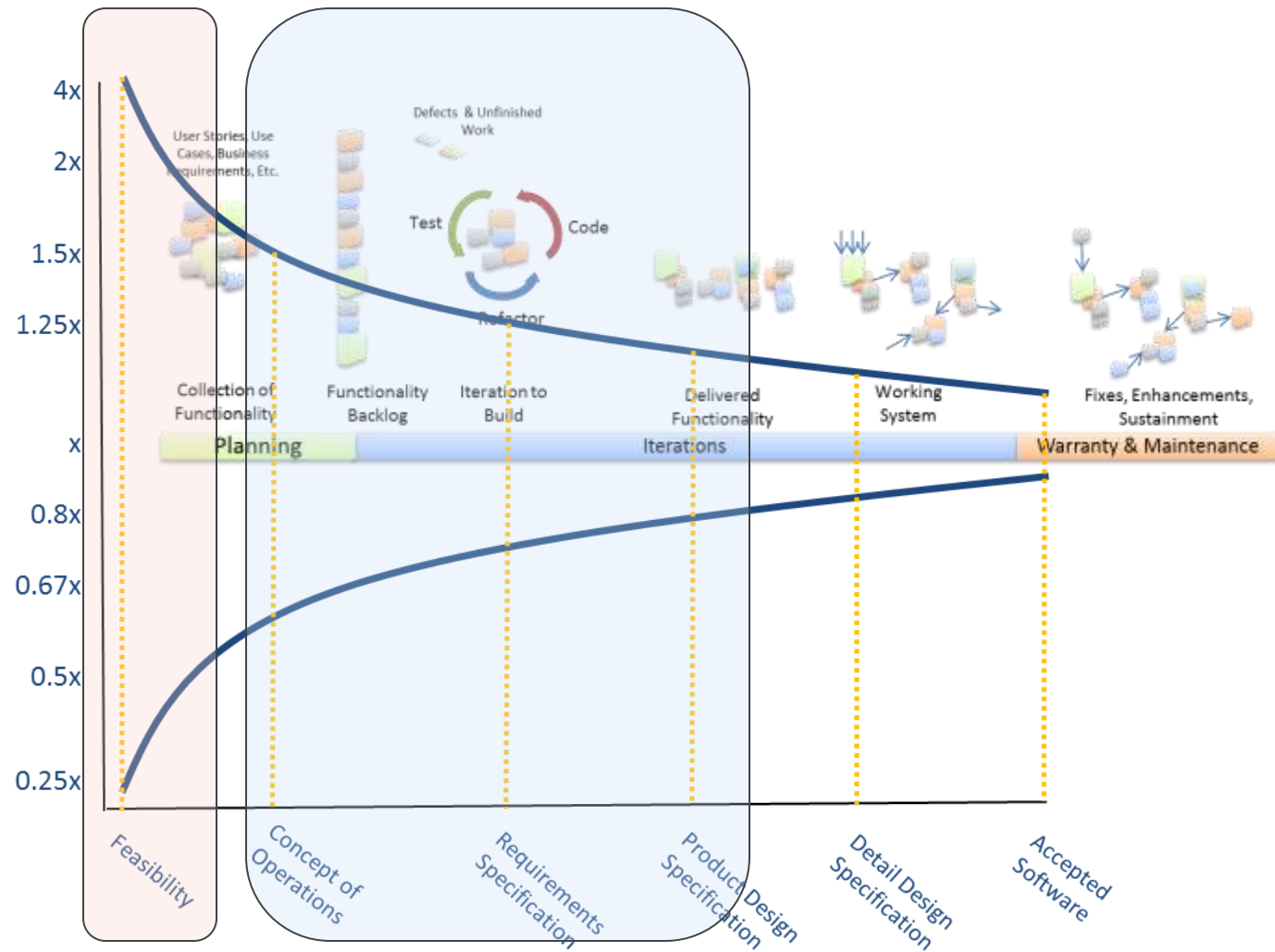


A classic case of the Flaw of Averages involves a statistician who drowns while crossing a river that is 3 ft. deep on average.

This poignant rendition by Jeff Danziger accompanied Dr. Savage's October 2000 article in the San Jose Mercury

Agile Uncertainty May Be The Same or Worse With Agile

- Precision comes over time! And what that it is unclear



Trouble Starts By Ignoring Project / Program Iron Triangle Realities

- Typical Trouble: Mandated features needed within specific time by given resources

Scope (features, functionality)

Resources

Quality

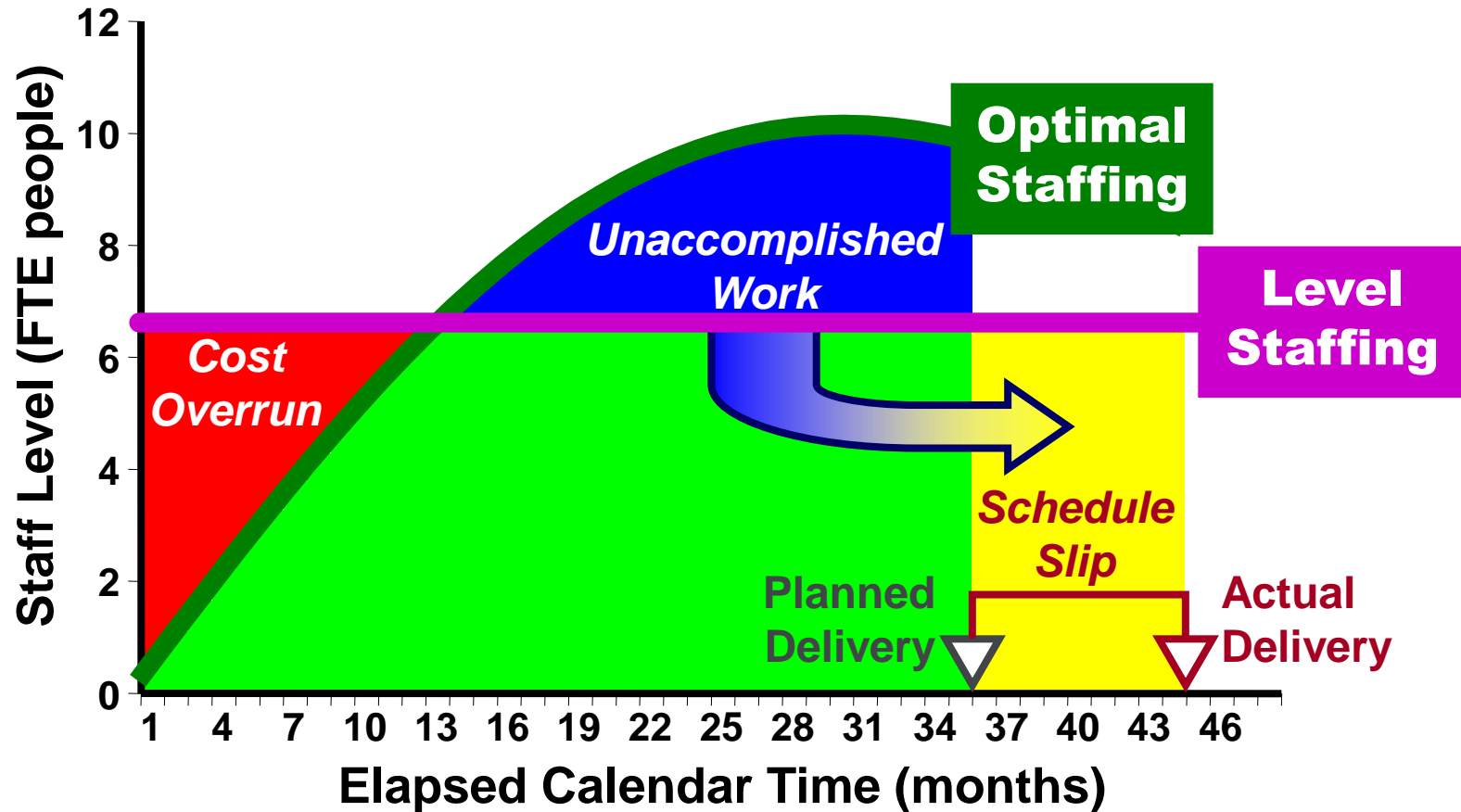
Schedule

- At least one must vary otherwise quality suffers and system may enter impossible zone!



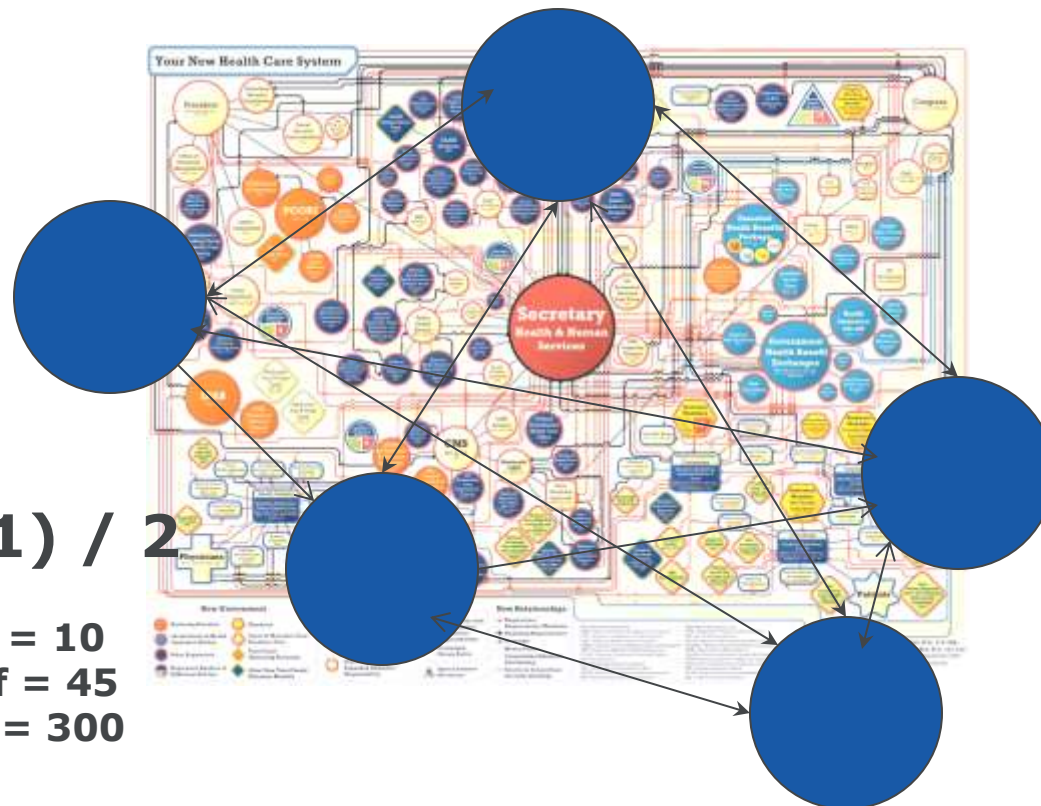
Pick Two

Avoid "Death Marches" and Failed Projects By Applying "Brooks Law"



Effective Staffing Staffing Beyond Plan Overstaffed Understaffed

Communications Are Challenging and Get Worse as Number of Organizations & Staff Increase



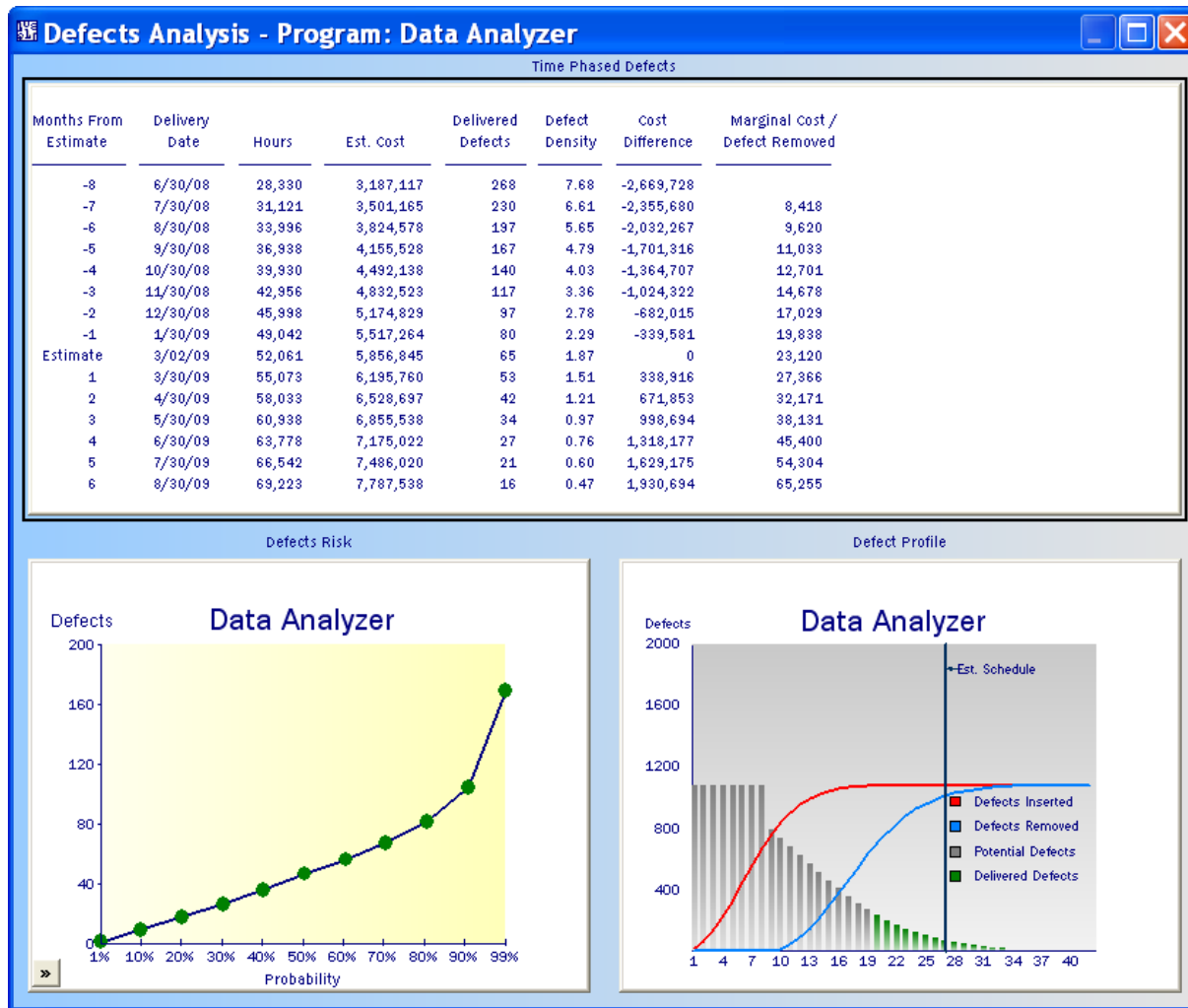
$$n(n - 1) / 2$$

5 Staff = 10
10 Staff = 45
25 Staff = 300

Why should we care: You can't usually make up schedule by adding staff

Problem Worse As Staff AND As Organizations Increase

Deploying Before Complete Leads To Program Disasters



“Brooks Law...but in this case it's irrelevant because the system was delivered on time -- it just didn't function correctly” Wrong: Shipping early doesn't mitigate Brooks Law

Shipping Early Is Disastrous

Defects Analysis - Program: Data Analyzer

Time Phased Defects

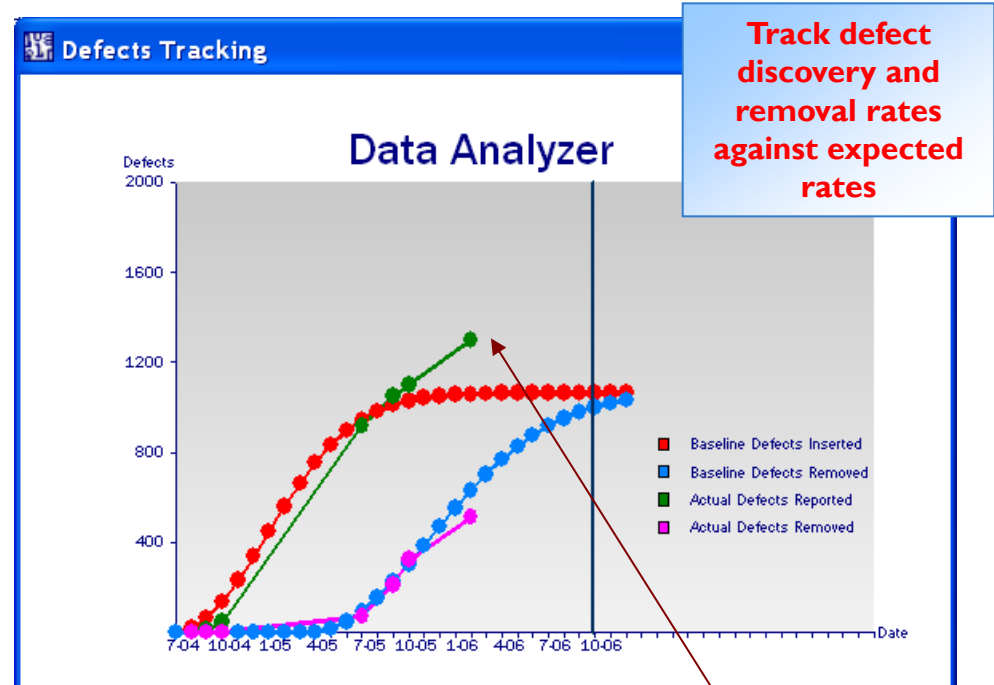
Months From Estimate	Delivery Date	Hours	Est. Cost	Delivered Defects	Defect Density	Cost	Marginal Cost /
-8	6/30/08	28,330	3,187,117	268			
-7	7/30/08	31,121	3,501,165	230			
-6	8/30/08	33,996	3,824,578	197	5.6		
-5	9/30/08	36,938	4,155,528	167	4.79	701,316	11,033
-4	10/30/08	39,930	4,492,138	140	4.03	1,364,707	12,701
-3	11/30/08	42,956	4,832,523	117	3.36	1,024,322	14,678
-2	12/30/08	45,998	5,174,829	97	2.78	682,015	17,029
-1	1/30/09	49,042	5,517,264		2.29	339,581	19,838
Estimate	3/02/09	52,061	5,856,845	65	1.87	0	23,120
1	3/30/09	55,073	6,195,760		1.51	338,916	27,366
2	4/30/09	58,033	6,528,697	42	1.21	671,853	32,171
3	5/30/09	60,938	6,855,538	34	0.8	998,694	38,131
4	6/30/09	63,778	7,175,022	27			
5	7/30/09	66,542	7,486,020	21			
6	8/30/09	69,223	7,787,538	16			

Example early ship shows 400%+ more defects Than recommended

Example deferred ship shows fewer defects. Can't get to zero

Software Progress and VIABLE SHIP DATE Can Be Determined By Defect Insertion & Removal

Health and Status Indicator
shows status and trends from
the previous snapshot
Thresholds are user definable



Health & Status Indicator

	Schedule Variance	Time Variance	Cost Variance	Size Growth	Defects
Analyst Support Sy...	BETTER	BETTER	WORSE	BETTER	WORSE

Increased defect
reporting rate
shows a
worsening trend

Packaged Applications Still Require Significant Testing

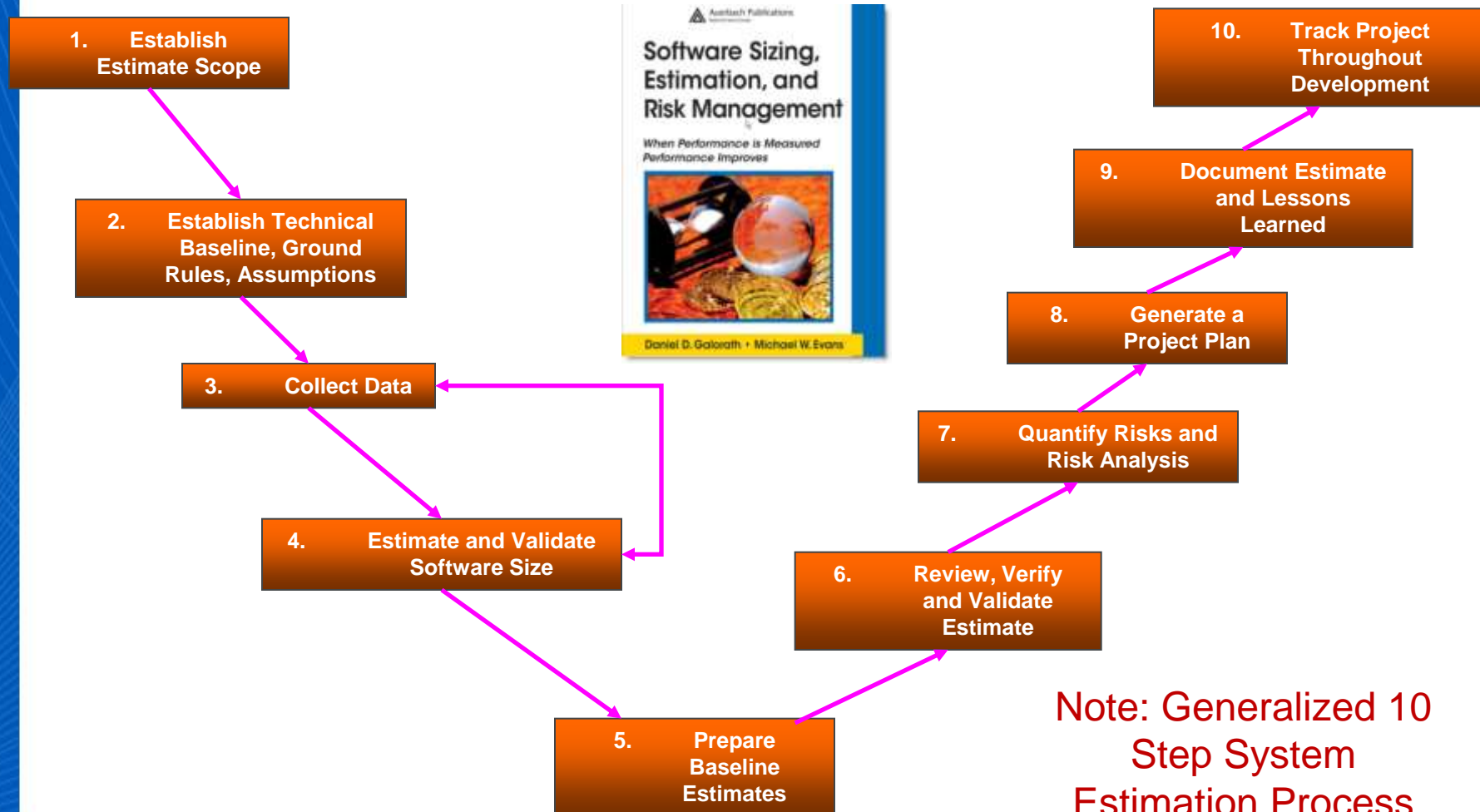


- Definition: "Commercial application program or collection of programs developed to meet needs of a variety of users, rather than custom designed for a specific organization"
- Many are enterprise applications
- Often allows / requires customization
- Examples: SAP; Rational PPM, SEER for Software; Microsoft Excel, CA Clarity, Oracle Business Suite

"One-third [of the budget] has to go to testing. Don't ever short change testing. Everyone always underestimates it, and says it's the last thing to worry about. Don't do that!"

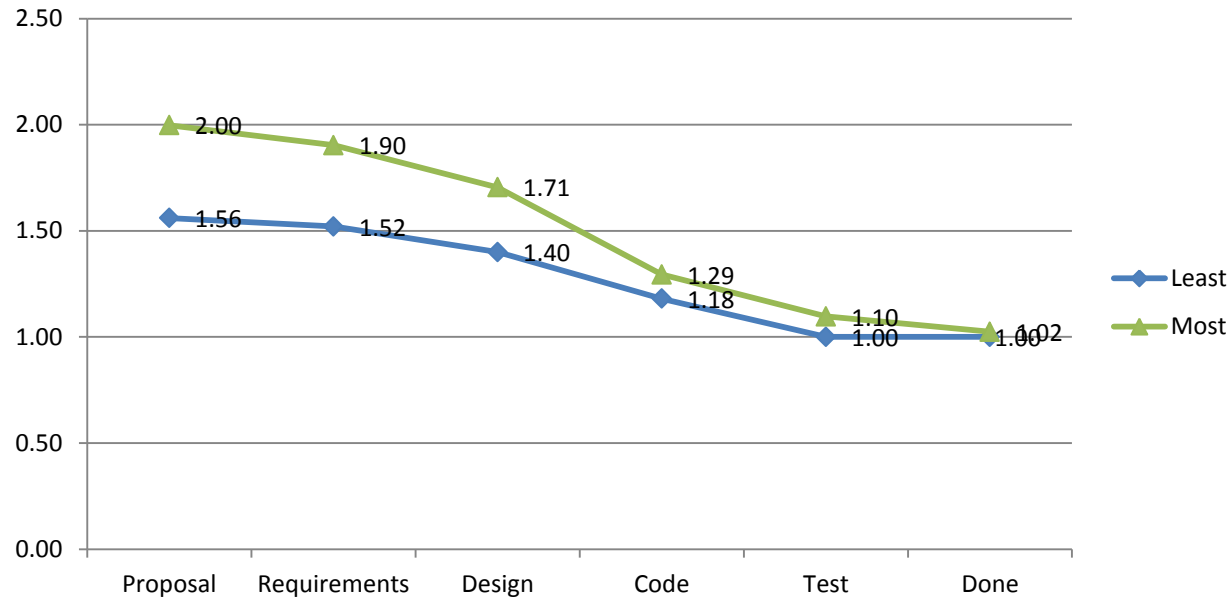
- Jim Larson, consultant for communications solutions provider

10 Step Software Estimation Process: Consistent Processes = Reliable Estimates = Successful Programs



Note: Generalized 10
Step System
Estimation Process
Also Available

Estimates and Plans Must Consider Functional Growth To Be Viable

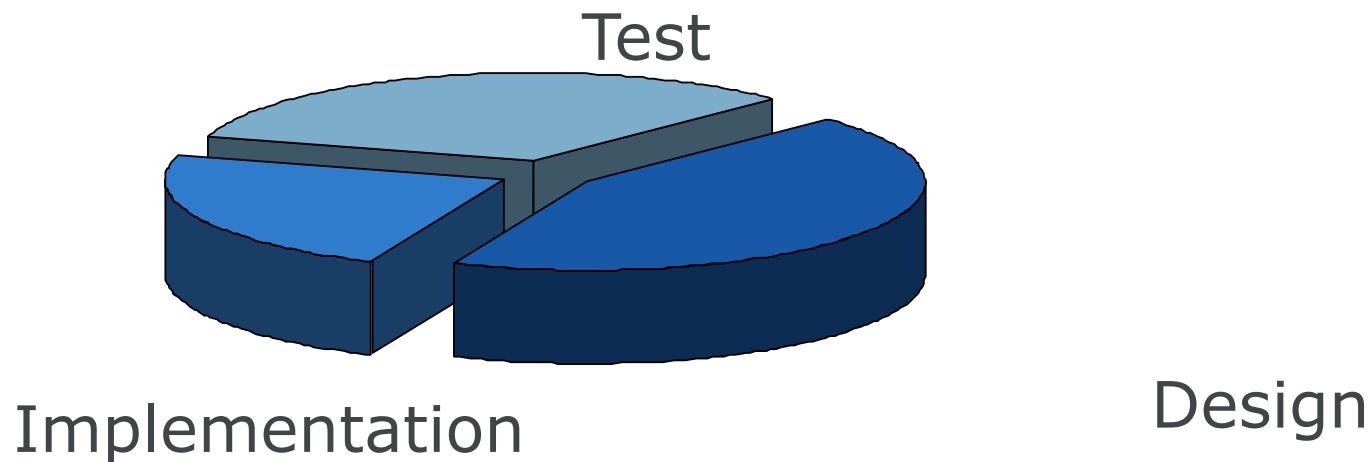


- Growth Range From Initial Sizing To Delivery
- Probable Growth is often early 2 to 1 for systems during early concept
- Many tools & Databases to estimate size (e.g. ISBSG)

Why should we care: If functional growth (requirements creep) not considered overruns are likely

Reuse: Watch Out For Low Cost Assumptions on "Heritage"

- Reuse or Heritage: applying existing software to a new mission (or additional innovation in its current mission)
- Effort to reuse software is routinely under estimated



Why should we care: Bad heritage assumptions often cause major schedule / cost overruns

IT Services Costs Must Consider Service Level Required

- High profile public system will have limited tolerance for down time
- Plan for equivalent of gold SLA when staffing operational support

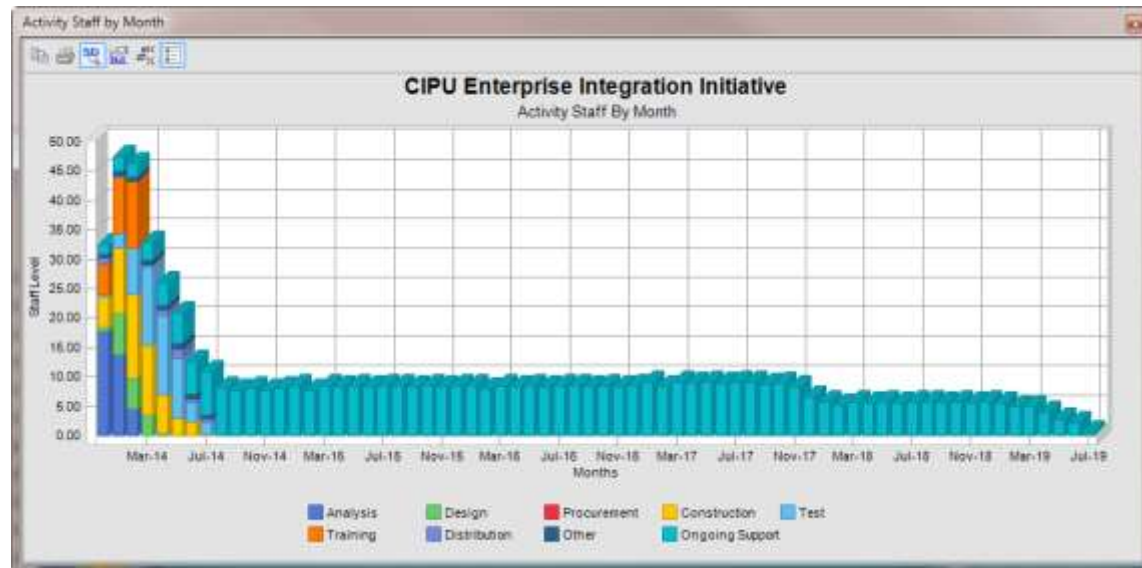
Service Level Agreement (Application Support)

Author: [Name] [Title] [Phone] [Email]

Service Level Target Benchmark for application support, often measured as a percentage of calls answered within a defined timeframe. Note that specific targets and SLA levels vary widely depending on the industry and the nature of the help desk.

Rating	Description
Very High	Acknowledged within 15 minutes and fixed within 2 hours (Gold).
High	Acknowledged within 1 hour and fixed within 4 hours.
Normal	Acknowledged within 4 hours and fixed within 8 hours (Silver).
Low	Acknowledged within 24 hours and fixed within 48 hours (Bronze).
Very Low	Acknowledged within 2 business days and fixed within 5 business days (Black).

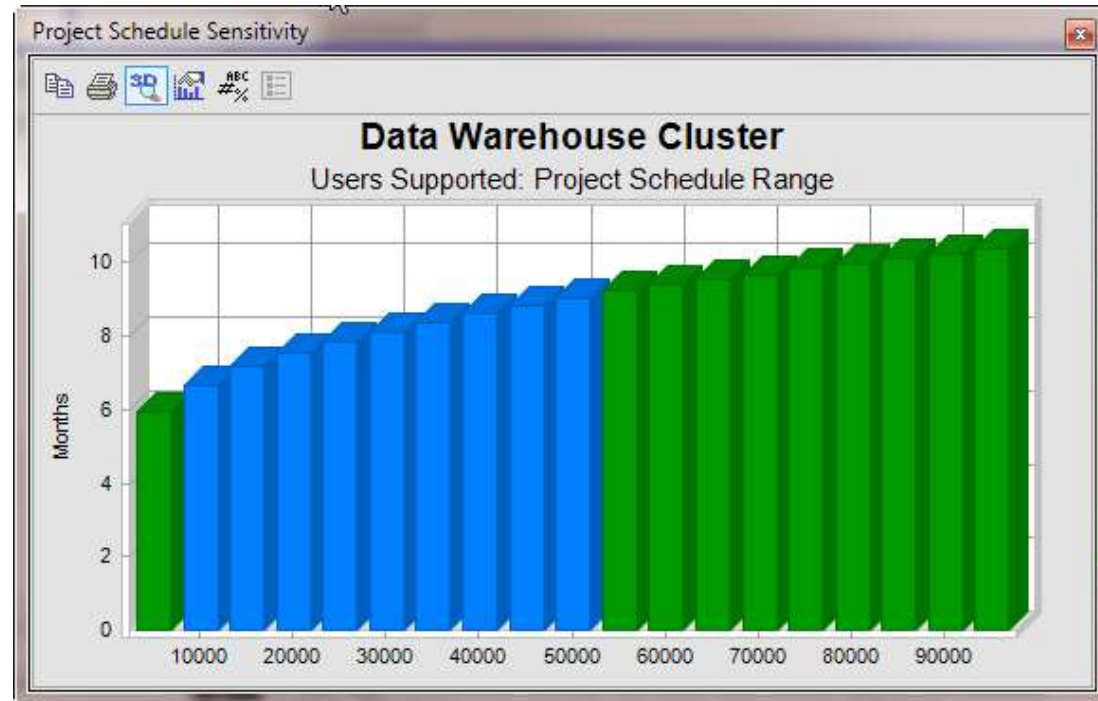
Each rating may be modified by a plus or minus to indicate actual ratings that are slightly higher or lower than what is indicated on the scale. For example, Normal+ would be slightly higher than Normal and Normal- would be slightly lower.



Up front testing needs more people.... Support must keep people ready to support users

Test In Production Environment To Avoid Surprises

- High profile new site can expect a surge of concurrent users
- Don't field without knowing concerns
- Anticipated concurrent users increase test time dramatically



Example shows nearly 40% additional test time going from 10k to 50k concurrent users

Minimal User Skill Increases Support Required

- Tier 1 support is inversely proportional to user training and skill
- Users will have no prior knowledge of system or procedures which will drive help desk staffing



Plan for this pain even
if the system runs perfectly

Software Implemented Security and Safety Requirements Add Significant Cost & Schedule



Security Requirements

Notes: Maximize Note...

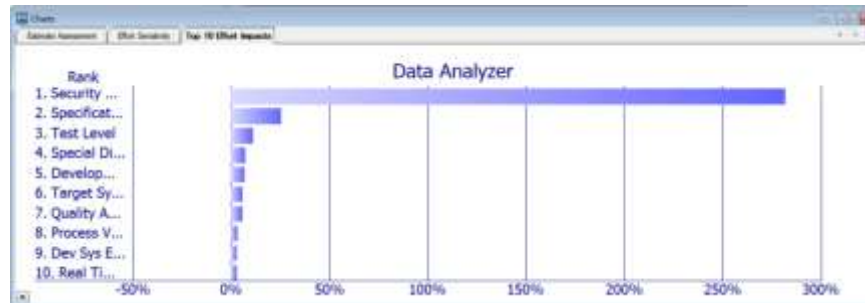
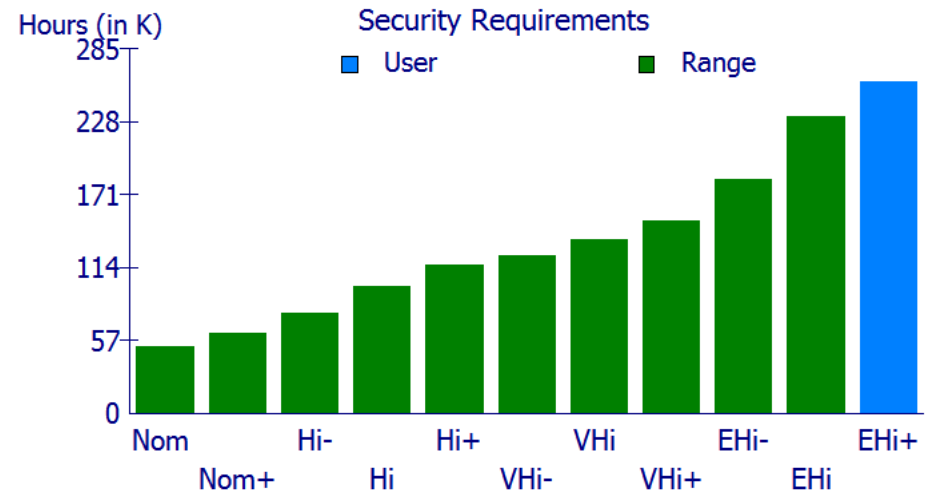
Least: EHi+ | Likely: EHi+ | Most: EHi+

OK, Cancel, Prev, Next, FBase, More Help

Security Requirements

Effort that will be expended to develop and certify security for this WBS item.

Rating	Description
Extra Hi+	Class A1: Security formally verified by mathematical proof. (Extremely rare). DO178B - Level A: Software whose anomalous behavior, as shown by the system safety assessment process, would cause or contribute to a failure of system function resulting in a catastrophic failure condition for the aircraft.
Extra Hi-	Common Criteria - EAL 7: Formally Verified Design and Tested. The formal model is supplemented by a formal presentation of the functional specification and high level design showing correspondence. Evidence of developer "white box" testing and complete independent confirmation of developer test results are required. Complexity of the design must be minimized.



Why should we care: Software implemented security and safety requirements can drive costs thru the roof

Large Systems Need Risk Analysis



- Both Schedule and Cost risk must be considered
- If every item in the plan is 90% probability the total project probability is much lower
 - $P(N \text{ elements Successful}) = (A_{\text{prob}}) (B_{\text{prob}}) \dots (N_{\text{prob}})$
 - For just 3 independent elements each with a 90% probability
 - $P(3 \text{ Elements Successful}) = (.9)(.9)(.9) = .729$
- For massive systems sophisticated risk analysis should be performed and dependencies considered
- Sophisticated (Monte Carlo Type) analysis should be used



Why should we care: Software & IT Systems are full of risks (and some opportunities)

Just a Single Point Usually Doesn't Reflect Reality (Adapted From SEI)

Process	Durations		
Step		Expected	
1		30	
2		50	
3		80	
4		50	
5		90	
6		25	
7		35	
8		45	
9		70	
10		25	
		500	

What would you forecast the schedule duration to be?

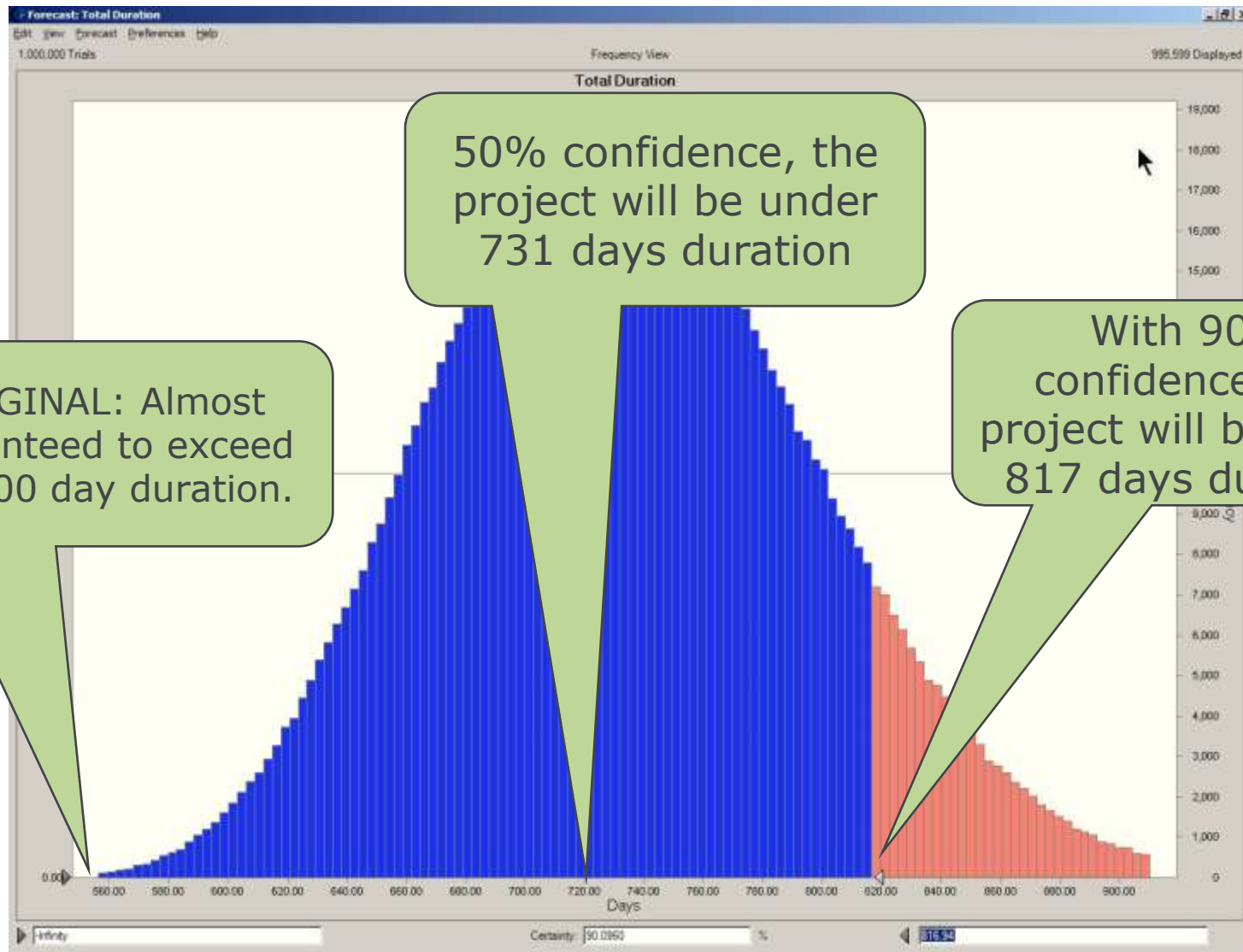
Range Clarifies Risk -2 (Adapted from SEI)

Process	Durations		
Step	Best	Expected	Worst
1	27	30	75
2	45	50	125
3	72	80	200
4	45	50	125
5	81	90	225
6	23	25	63
7	32	35	88
8	41	45	113
9	63	70	175
10	23	25	63
	452	500	1252

What would you forecast the schedule duration to be now?

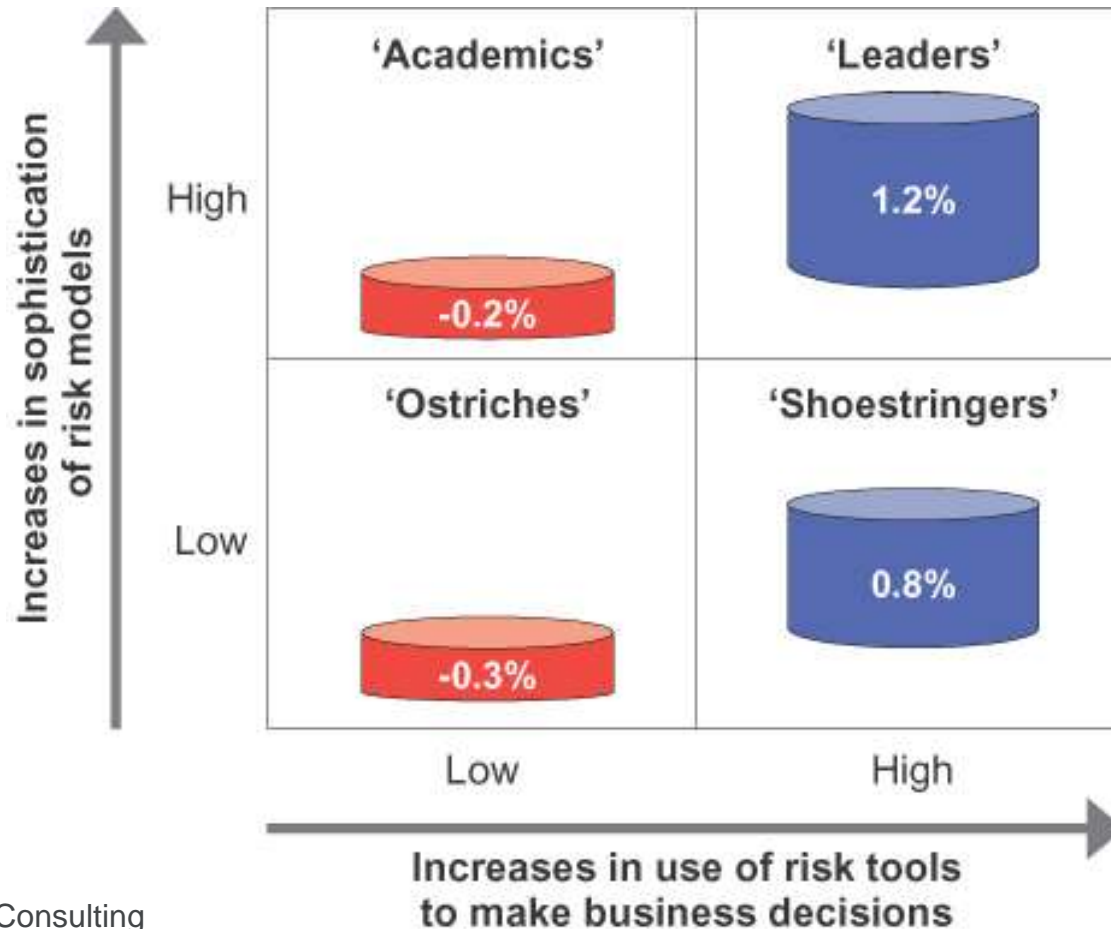
Capture of uncertainty is a major improvement

Risk Analysis Makes Projects More Successful – 3 (Adapted from SEI)



Managing Risk Improves Results

- Annualized total shareholder returns (1998-2003) for differing degrees of risk model sophistication and risk tool usage



Source: PA Consulting
Survey of Global Banks

Step 9 Assess Alternatives & Select

Step 9. Assess
Alternatives &
Select Optimal
Alternative

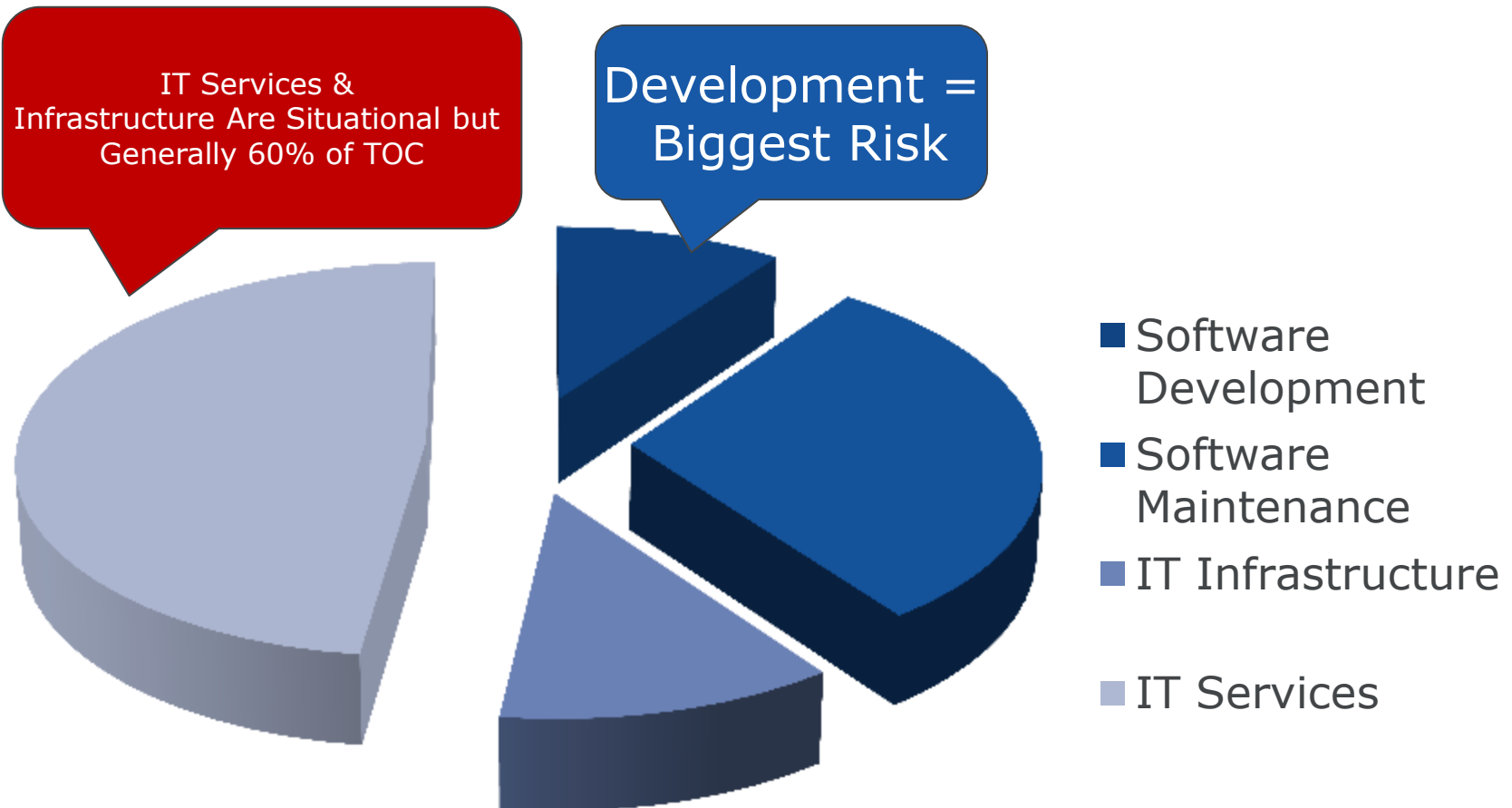
- Use the figures of merit to determine which is the best
 - Lowest risk
 - Highest value
 - Scored Weighted importance

Weighted Rating evaluation Example (Source: Acedemia.edu)

		Concept Alternatives					
		gears		v-belts		chain	
Criteria	Importance Weight (%)	Rating	Weighted Rating	Rating	Weighted Rating	Rating	Weighted Rating
high efficiency	30	4	1.20	2	0.60	3	0.90
high reliability	25	4	1.00	3	0.75	3	0.75
low maintenance	20	4	0.80	3	0.60	2	0.40
low cost	15	2	0.30	4	0.60	3	0.45
light weight	10	2	0.20	4	0.40	3	0.30
	100	NA	3.50	NA	2.95	NA	2.80

Rating	Value
Unsatisfactory	0
Just tolerable	1
Adequate	2
Good	3
Very Good	4

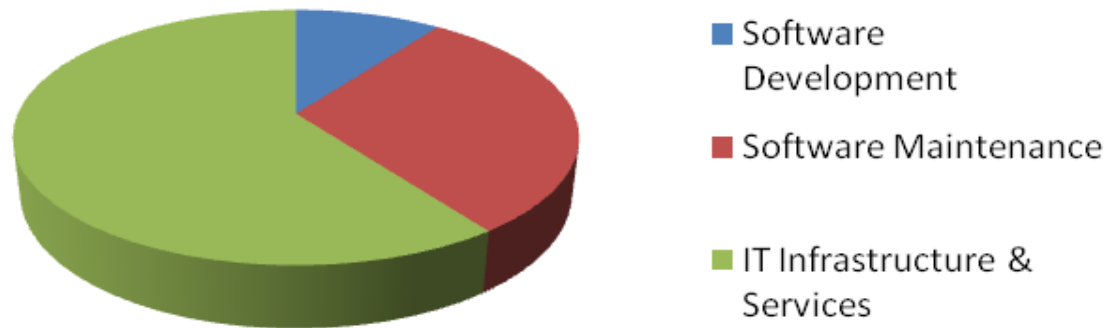
Example: Traditional On Premises Software Total Ownership Cost Allocation



Software Development is about 6-10% of total ownership cost...
But much more of the risk
Assume \$10m development could be over \$100m total ownership

Evaluate Total Ownership Costs, Not Just Developments: IT Systems Total Ownership Costs; 60+% Can Be Infrastructure & Services

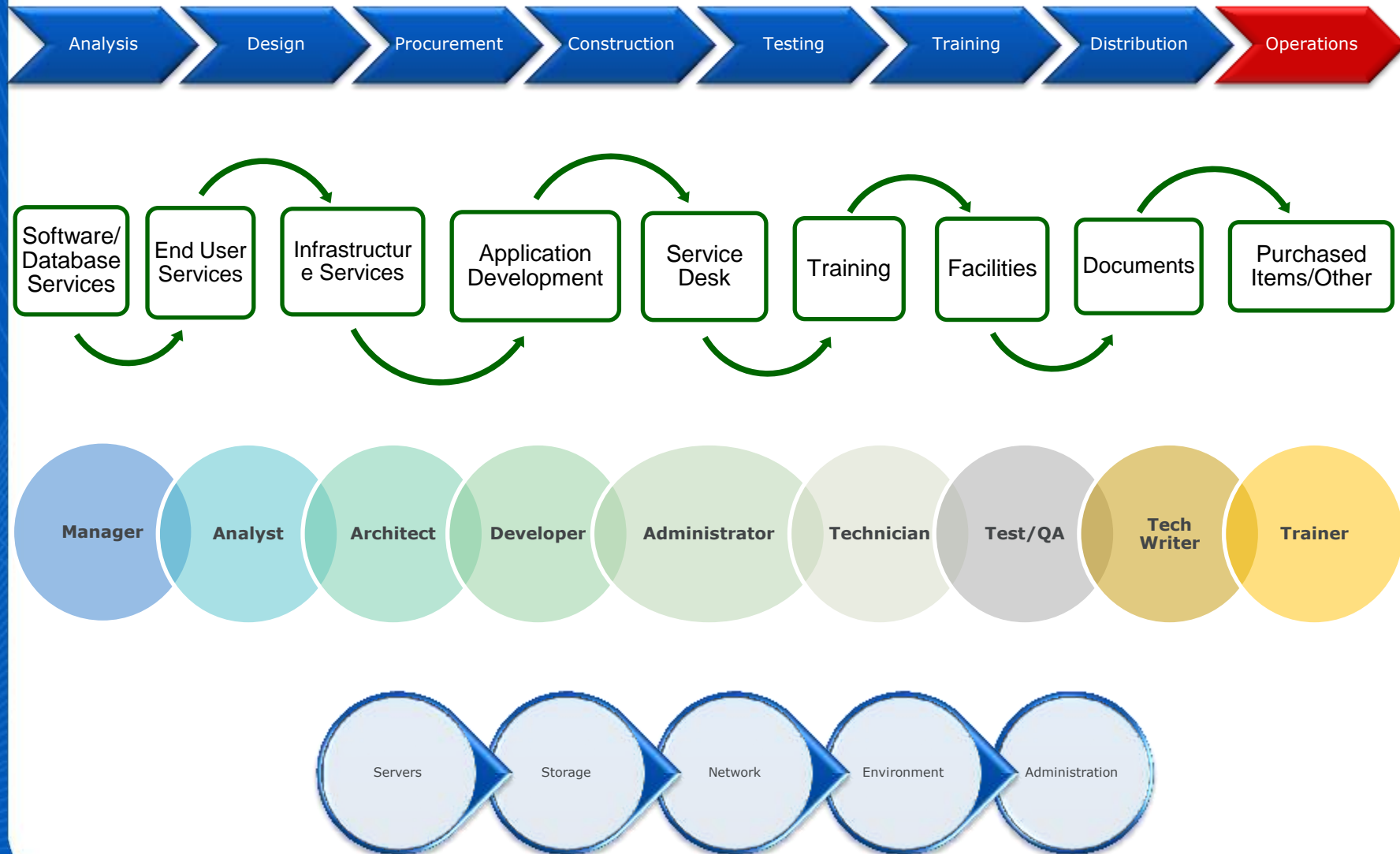
Total Ownership Cost: Typical Relative Cost By Activity



Software Development is about 6-10% of total ownership cost...But much more of the risk

Assume \$10m development could be over \$100m total ownership... But it must be done

Cloud Example: Labor & Hardware Change From Iaas To PaaS To SaaS

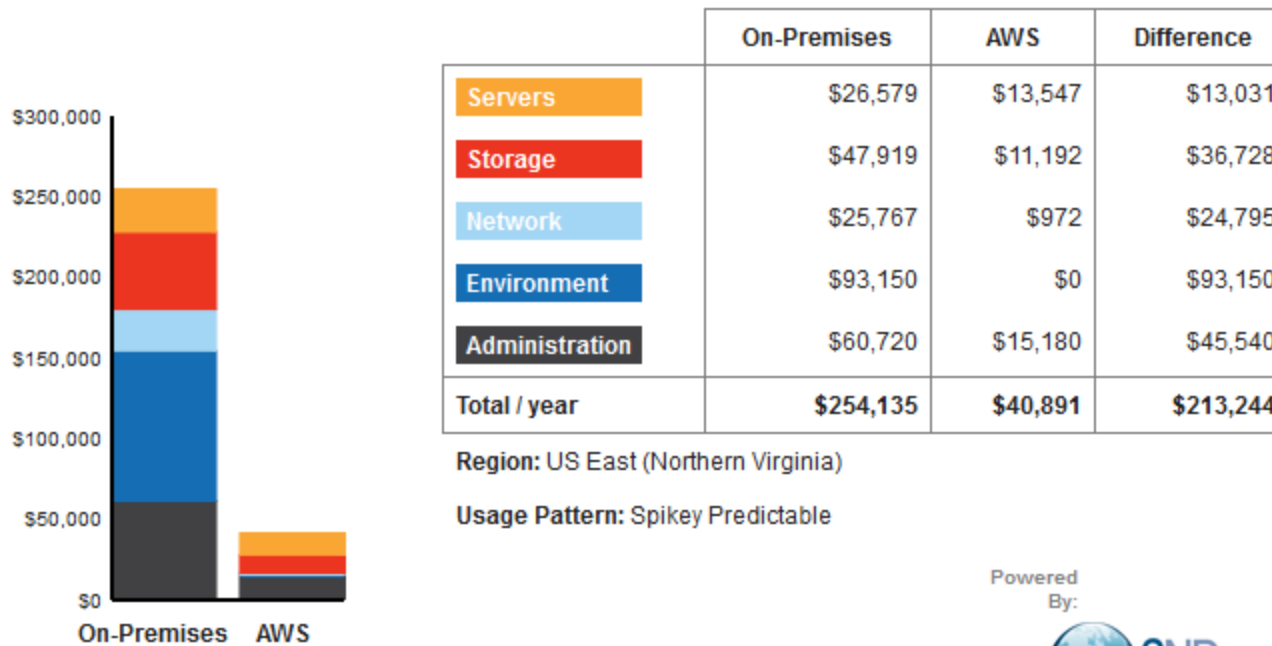


Cloud Example: Current Costs of IaaS Are Readily Available

TCO Comparison Calculator for Web Applications* (Beta)

[<< Adjust Calculator Settings](#)

You could save **\$213,244** per year running on AWS.



<http://tco.2ndwatch.com/#compare>

Step 10 Document Analysis and Lessons learned

Step 10. Document Analysis and Lessons Learned

- Document estimate complete AND project complete
- Lessons learned ASAP while memories are still fresh
 - Provides evidence that your process was valid
 - Can substantiate or calibrate your estimation models
 - Provides opportunity to improve estimating process
- Missing or incomplete information & risks, issues, and problems the process addressed & any complications that arose
- Key decisions made during the estimate & results
- Dynamics that occurred during the process e.g.
 - Interactions of your estimation team
 - Interfaces with clients
 - Trade-offs made to address issues during the process

Conclusions: IT Systems Are Hard

- Healthcare.gov Environment Was difficult
 - Requirements Volatility
 - Complexity
 - Extensive integration
 - Legacy systems
 - Forced deadline
- Lessons learned yet again
 - Maintain strong & enabled leadership... Executives need viable information
 - Communicate constantly and completely at all levels
 - Iron triangle rules: Keep requirements stable or... defer features... to keep the date
 - Include risk in plans and Practice extensive risk management
 - Test early, often and end to end
 - Don't just blame the developers
 - Use commercial off the shelf when possible when viable

**Estimation, planning, control can help but....
Mandate the possible**

Key Points

Viable
affordability
decisions yield
project
achievements



Repeatable
affordability
process is a
key method
of analyzing
affordability

We can make
best value
decisions,
driving down
cost & increasing
value



estimate

estimate • analyze • plan • control

Estimation Best Practices



- Decide Why You Want An Estimate
- Map Estimation Goals To Estimate Process Maturity & Develop Plan To Achieve The Maturity
- Have A Documented, Repeatable Estimation Process
- Make The Estimating Process As Simple As Possible; But No Simpler
- Be Proactive: The Process Is Important, The Tools Go Along With The Process
- Get Buy-in From Program Managers
- Hold People Accountable: Center Of Excellence Can Prepare Estimate But Program Managers Must Own Them
- Tie The Estimate To The Plan

Estimation Best Practices 2

- Evaluate Total Ownership Cost; Not Just Development
- Estimate A Range And Pick A Point For The Plan
- Re-estimate The Program When It Changes
- Avoid Death Marches: Programs With Unachievable Schedules Are Likely To Fail And Drain Morale
- Keep A History: Start An Enterprise Database NOW...
- Business Case: Evaluate ROI In Addition To Costs
- Convert Expert Spreadsheets Into A Common Language



Estimation Best Practices 3

- Track Progress Vs. Estimate Throughout The Life Cycle
- Estimate Schedule As Well As Effort (Cost) For Complete Picture
- Tie The Business Case Into The Estimating Process
- Attack Non-productive Rework As Part Of The Process



Estimation Best Practices 4

- Have clear definitions:
 - What does “complete” mean
 - What activities are included and excluded (E.g. development only or total ownership; help desk included or excluded, etc.)
 - Which labor categories are included and excluded in the estimate (e.g. are managers included? Help desk? Etc.)
- Don't ignore IT infrastructure and IT services costs
- Tracking defect sources can go along with the process

estimate

estimate • analyze • plan • control

Backup Slides

