

estimate

estimate • analyze • plan • control

Planning and Managing Modernization Projects with Fewer Resources/Budget

IT Confidence Beijing



INTEGRITY INNOVATION EXCELLENCE

Key Points

Modernization
can be risky &
substantially
misestimated



Modernization
approaches can
reduce cost &
risk

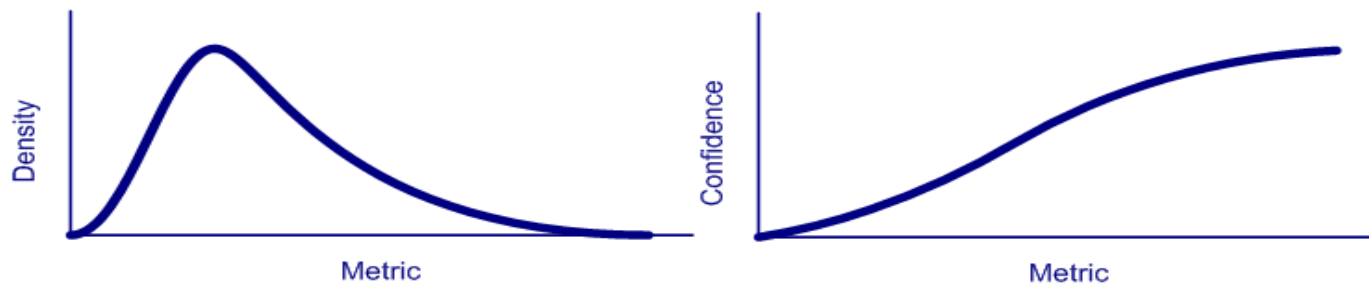


Open source,
Agile, Cloud &
other technologies
can help... But
they come at a cost
(not free)



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***



Human Nature: YOUR PEOPLE Are Optimism Biased



Harvard Business Review explains this Nobel Prize Winning Phenomenon:

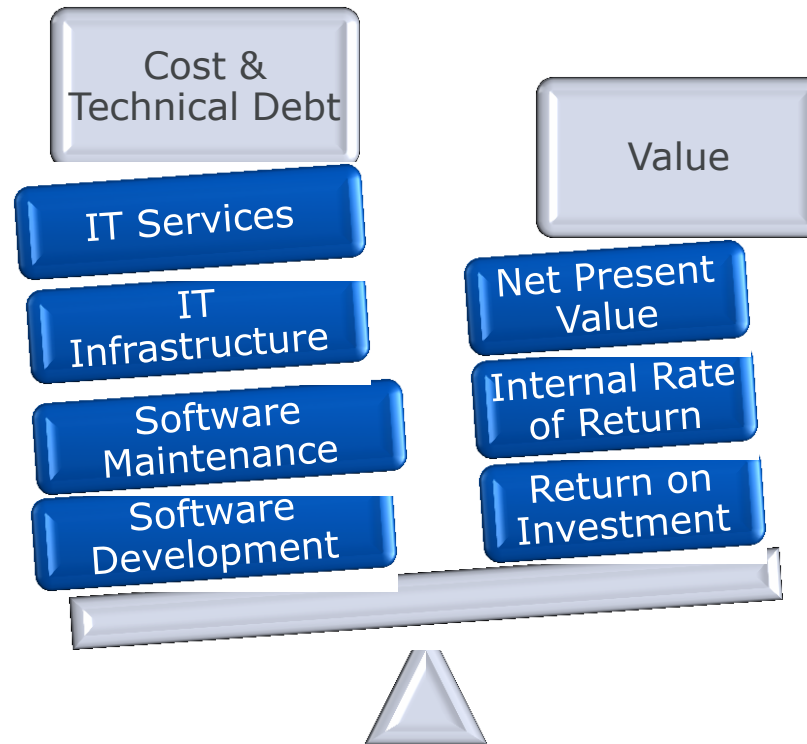
- Humans seem hardwired to be optimists
- Routinely exaggerate benefits and discount costs
- Bias permeates opinions & decisions & causes waste & failure

Delusions of Success: How Optimism Undermines Executives' Decisions (Source: HBR Articles | [Dan Lovallo](#), [Daniel Kahneman](#) | Jul 01, 2003)

**Solution - Temper with “outside view”:
Past measurement results, traditional forecasting, risk
analysis and statistical parametrics can help**

**Don't remove optimism, but balance optimism and
realism**

Best Analysis of Modernization Approach Looks at Value & Time To Value to the Business



“Economics is primarily a science of choice
software economics should provide methods for analyzing the
choices software projects must make.” Leon Levy

Cost & Technical R's of modernization

(Adapted from Microsoft & Gartner)

Retire

- Decommission if legacy app providing little value
- Possibly roll some legacy functionality into consolidated modern application

Refactor

- Preserve behavior by improving existing code
- Possibly execute on new infrastructure (PaaS)

Replace

- If legacy app providing value but commercial alternative can be better

Retain & Wrap

- RETAIN if inexpensive or impractical to modernize
- WRAP: modern wrapper around app - additional value & benefits e.g. C# Java wrapper around COBOL app

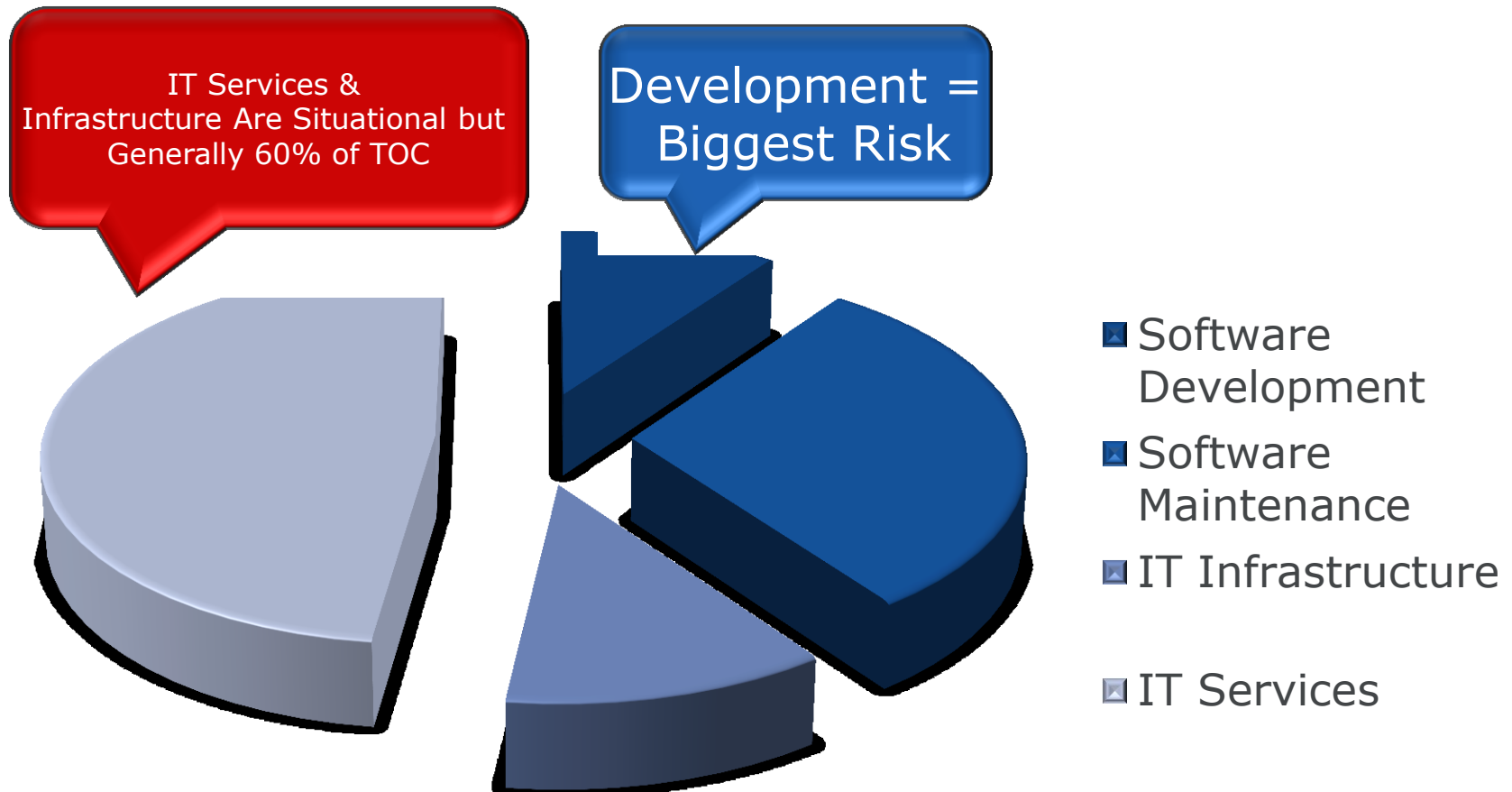
Rehost

- Viable functionality but Expensive to run
- Move VM from on-premises to new environment E.g. IaaS

Redevelop

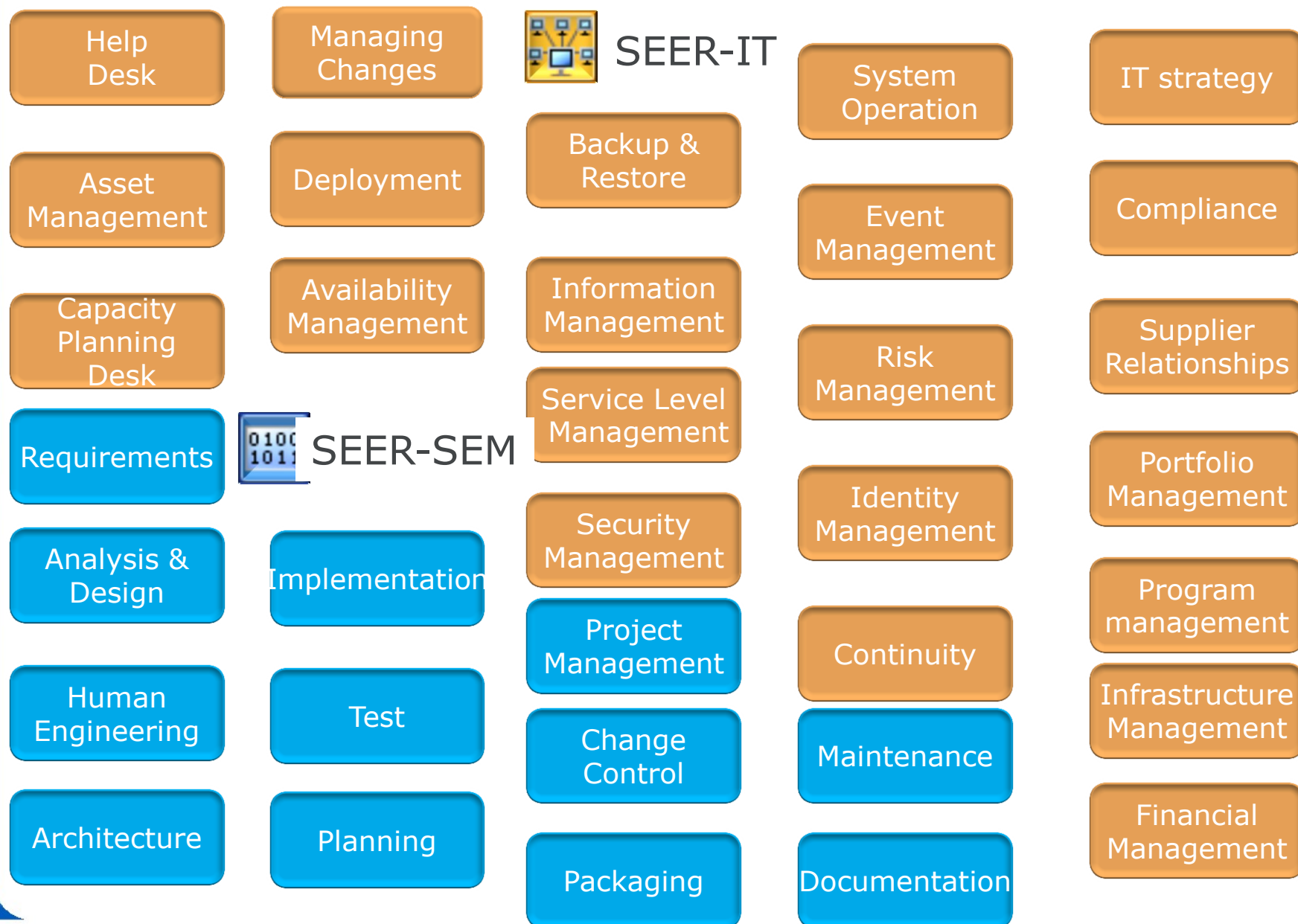
- Application providing value but legacy language, environment
- Rewrite a new application that meets the current and upcoming requirements

Example: Software Total Ownership Cost Allocation



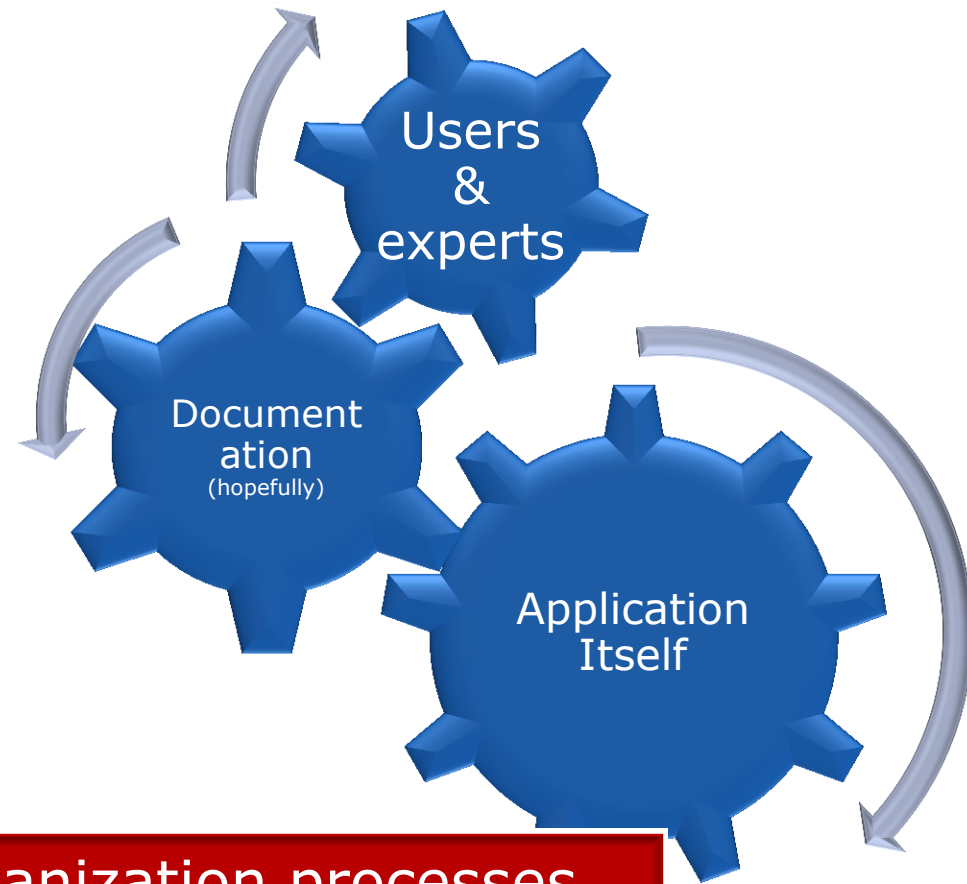
Software Development is about 6-10% of total ownership cost...
But much more of the risk
Assume \$10m development could be over \$100m total ownership

Modernization & Value: Software & IT Should Both Be Estimated (Adapted from IBM)



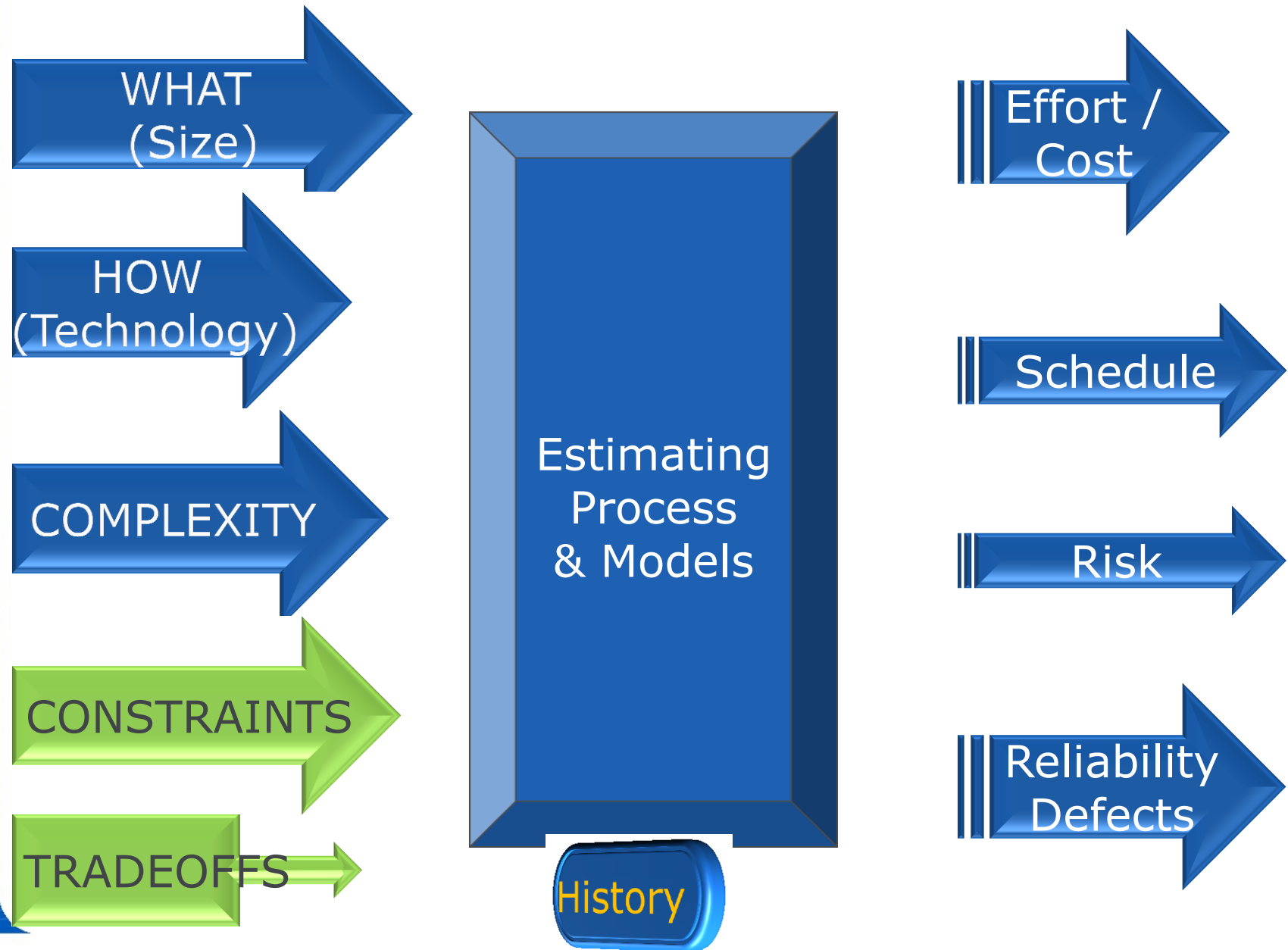
Modernization Requires “As Is” Model Discovery

- “As Is” usually requires discovery to mine knowledge
 - Business processes
 - Business rules & vocabulary
 - Logical data model models
 - Application logic
 - Physical data model
 - Program logic



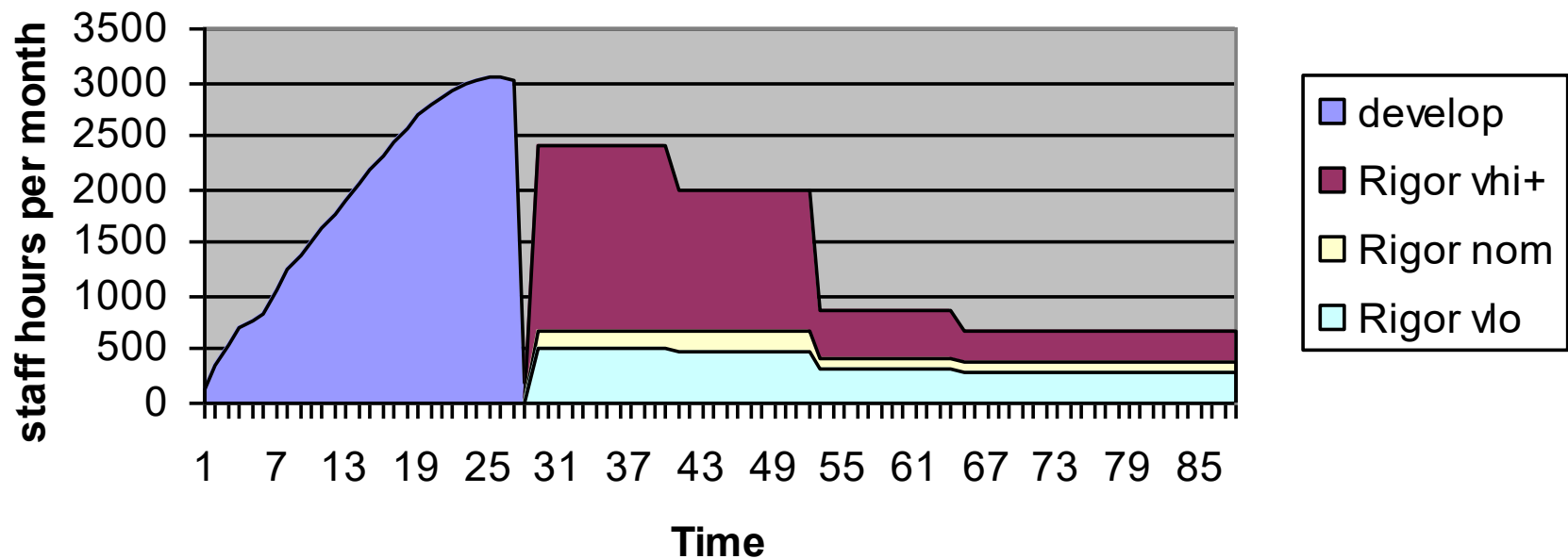
Trying to change the organization processes just because of new software can be disaster

Mining Structural Quality Metrics For Existing Software Helps Estimate Modernization



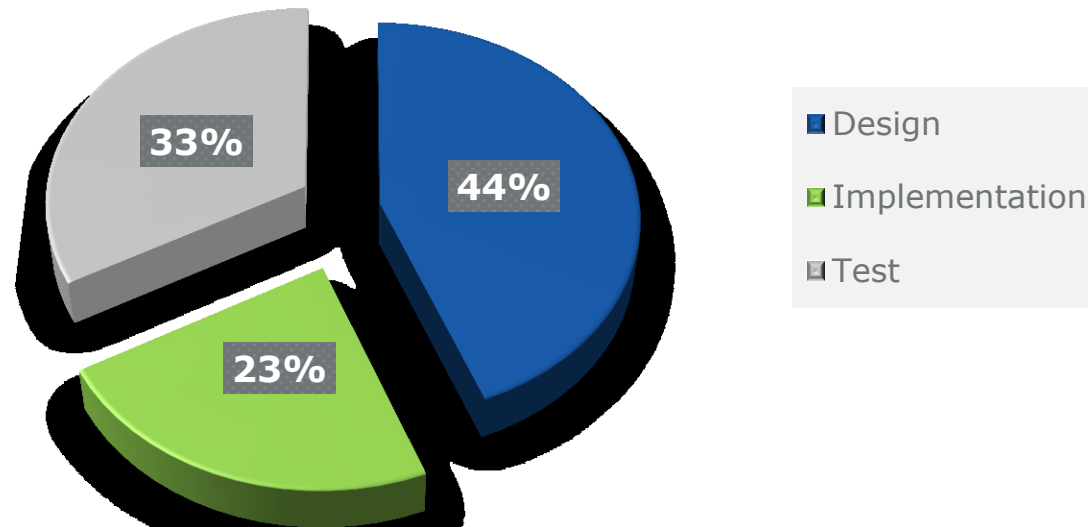
Legacy Systems Have Substantial Costs That Modernization May Offset

Staff Vs Maintenance Rigor



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

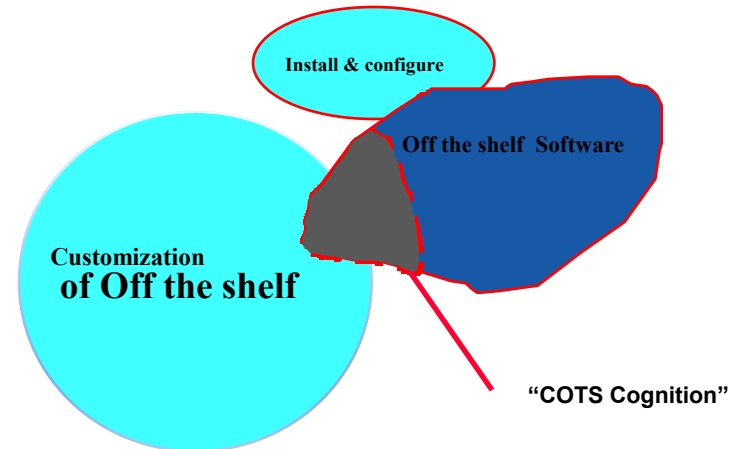
Key Components Of A Software Project That Uses Off the Shelf

- Developmental Software:

- Functionality developed specifically for the project at hand
- May include customization of COTS

- “Glue” Code:

- Code written to bind COTS to developmental software
- Development effort must be captured



- Install & configure

- COTS Software:

- Purchased functionality
 - **Direct Cost** component of COTS integration

- COTS Cognition:

- Required functionality within the COTS software that must be understood
 - **Effort** component of COTS integration

DevOps (Demand Model Can Control & Reduce Costs) Adapted from The Phoenix Project



Project Work

- Projects the business is demanding
- What business is asking development or IT for
- Usually managed in some way

Internal Work

- Housekeeping: Any internally focused activity
- E.g. installing devices, decommissioning datacenter
- Usually with little oversight or visibility and consume untold resources

Operational Changes

- Every day IT operations
- Planning, assessing, building, testing and deploying changes
- May include managing the for above

Unplanned Work

- Major source of IT technical debt
- 25% – 40% of all spending on projects is wasted as a result of re-work (Carnegie Mellon)
- Recovery work: Can put everything else on the backburner,
- Usually takes you away from meeting your goals

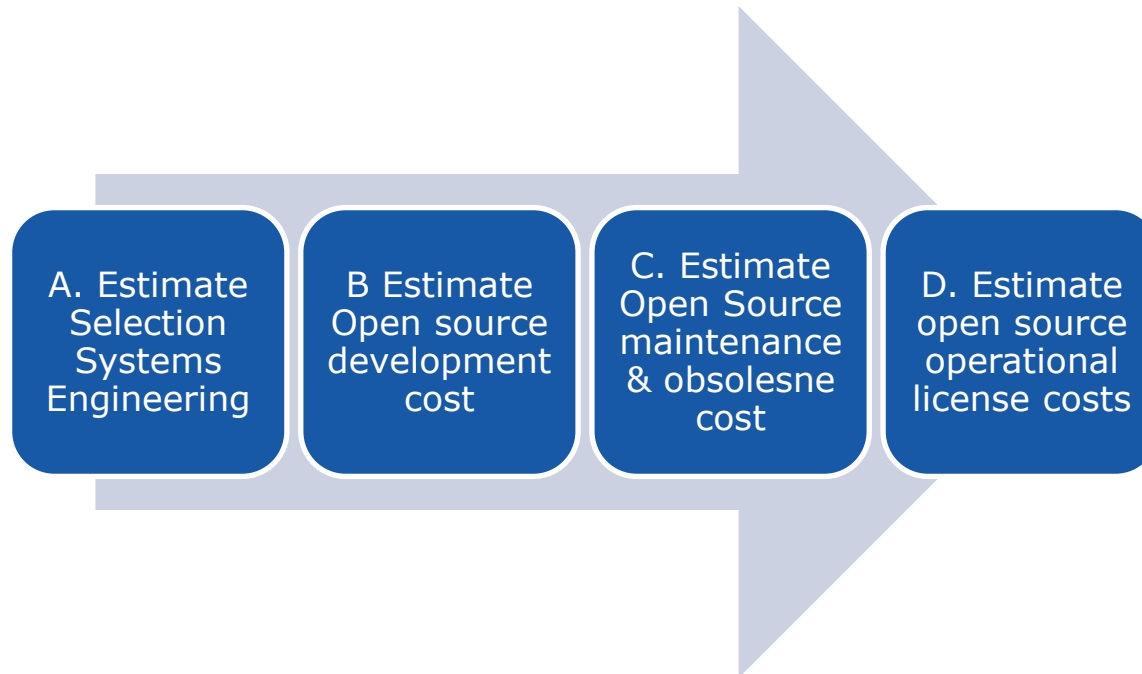
estimate

estimate • analyze • plan • control

Open Source



Estimate Open Source Costs



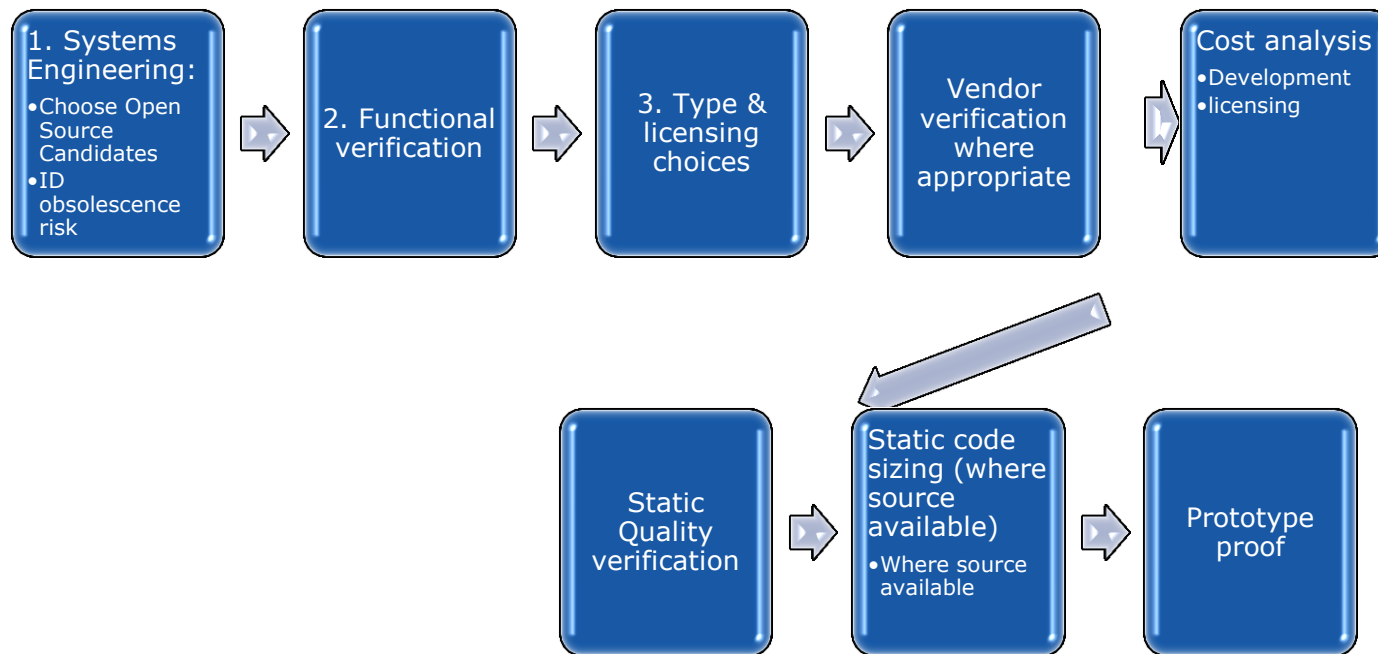
Different Open Source Acquisition Alternatives

Open Source Note	
Open Source Open Use	<ul style="list-style-type: none">• Source code, generally no warranty
Open Source Black Box Use	<ul style="list-style-type: none">• Can't be changed by organization• May be add-it capability
Open Source Black Box Use from Vendor	<ul style="list-style-type: none">• Can't be changed by organization• Vendor supports & updates
Open Source Open Use Developmental	<ul style="list-style-type: none">• Source code changed by organization• You own it, you maintain it• Updates from open source community will not be applicable or require individual integration

Open Source Summarized Costing Process

	X.1 Systems Engineering	X.2 Development	X.3 Maintenance	X.4 Additional Costs
Open Use	Compute Effective Size, Functionality or SLOC, or use Systems Engineering model	Use Effective Size	Cost Model with Use Total or Effective Size	Licensing Cost
Black Box Use	Compute Effective Size, Functionality or SLOC	Similar to Open Source Open Use	Same as Open Use	Licensing Cost
Black Box Use from Vendor	Compute Effective Size, Functionality or SLOC	Various, good approach is function points	Same as Open Use	Licensing Cost plus Support
Open Use Developmental	Compute Total, Effective, New Size	Estimate as Development	Same as Open Use	May have licensing cost

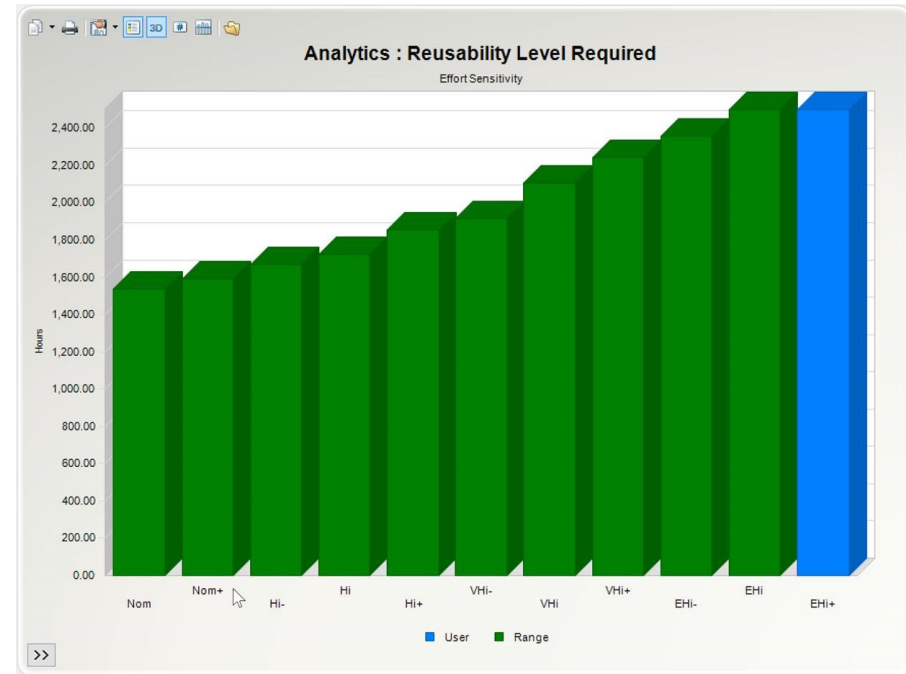
Open Source Selection Process



When source available static analysis can provide quality and size indications

US OMB M-16-21 Promote Reuse & Open Source

- [M-16-21](#), OMB's Federal Source Code Policy: Achieving Efficiency, Transparency, and Innovation through Reusable and Open Source Software requirements
- (1) all custom-developed code must be available for reuse within the government subject to limited exceptions (e.g., national security) and
- (2) under a three-year pilot program, federal agencies must release at least 20 percent of their custom-developed code to the public as OSS.
- Goal is to promote reuse as a cost saving measure to reduce redundant coding



- Problem: Up to 63% increase in initial development effort to make software reusable in the first place

Static Code Analysis Can Help Quantify Open Source Quality (Source Cast Software)



- **Reliable measurement:** CISQ Software Sizing and Quality Standards.



- **Automated:** Sizing AFP and AEFP by a tool which remove subjectivity.



- **Consistent:** Same rules and assumption from version to version.



- **Business relevant:** Risk adjusted Productivity with normalization for trending.



- **Fact based measurement:** All metrics quality, quality or complexity should be accessible by both side (client and vendor).



- **SLA or KPI:** All metrics quality, quality or complexity can be reuse in some contract focus on the evolution.

Coverity and Open Source Projects



- Coverity is providing a free service for open source projects



741 projects
2.5M LOC



44,641 defects are fixed

(Only 10.2% of identified defects are false positives in 2013)

Open Source Licenses Can Be A Cost Risk

synopsys.com/software-integrity/2016/10/07/five-software-licenses-you-need-to-understand/



Public Domain

- Anyone can do anything
- Doesn't mean it is safe

Permissive

- Minimal requirements on software modification or redistribution
- AKA: Apache Style or BSD Style or MIT license

LGPL

- GNU Lesser General Public License
- Any user must be given the right to modify so your developed code might have to be exposed

Copyleft

- End user can modify and distribute new works based on your work
- Derived works reside under the same license

Proprietary

- All rights reserved
- Software may not be modified or redistributed

Open Source Classification & Estimation Approach



- If used as is
Unchanged Non-Developmental Software (NDI) need
 - Selection
 - COTS Cognition
 - Integration & Test
 - Maintenance
- If customized needs to be estimated as developmental software
- BEWARE the cost of Government Furnished Data (GFD) open source.. May not be well suited to new application

Availability does not guarantee suitability, reliability, or information assurance

Estimating Non Developmental Open Source

1. If source available run code counter or automated function point sizing
 - Establish effective size
 - Use a cost model like SEER-SEM estimate the OSS effort & RISK to understand and maintain
2. If source code not available
 - Use function point analysis to count / estimate the effort involved in comprehending, applying, and testing the OSS.
 - Use a cost model like SEER-SEM that supports function point estimating and RISK for development and maintenance
 - Or Use Galorath COTS cognition to understand how much needs to be understood, used and tested
3. Estimate support costs from vendors
4. For OSS obsolescence

Assume how many years the OSS will remain viable... then use size measures from initial sizing to cost replacement + process to estimate rework to surrounding existing systems

Costing Open Source Source System Engineering

1. Identify requirement
2. Determine acceptable open source licensing alternatives (include compatibility with your PaaS if applicable)
3. Select alternatives
4. Evaluate viable alternatives
 - Technically
 - Licenses and support cost
 - Computing resources (if major component)
 - Obsolescence risk wise

Remember: Free in Open Source doesn't necessarily mean no cost

How To Compute Effective Size For Open Source

Step 1: Set Redesign Factors

Redesign Breakdown

Formula

$$0.22*A + 0.78*B + 0.5*C + 0.3*(1 - (0.22*A + 0.78*B)) * (3*D + E) / 4$$

Result Redesign Percentage

0.00% 0.00% 0.00%

Weight Redesign Component

Weight	Redesign Component
0.22	Architectural Design Change
0.78	Detailed Design Change
0.5	Reverse Engineering Required
0.225	Redocumentation Required
0.075	Revalidation Required

Least	Likely	Most	Percentage of the existing software that...
0%	0%	0%	... requires architectural design change
0%	0%	0%	... requires detailed design change
0%	0%	0%	... requires reverse engineering
0%	0%	0%	... requires redocumentation
0%	0%	0%	... requires revalidation with the new design

Step 2: Set Reimplementation Factors

Reimplementation Breakdown

Formula

$$.37*A + .11*B + .52*C$$

Result Reimplementation Percentage

0.00% 0.00% 0.00%

Weight Inputs

Weight	Inputs
0.37	Recoding Required
0.11	Code Review Required
0.52	Unit Testing Required

Least	Likely	Most	Percentage of the existing software that...
0%	0%	0%	... requires actual code changes
0%	0%	0%	... requires code reviews
0%	0%	0%	... requires unit testing

Step 3: Set Retest Factors

Retest Breakdown

Formula

$$.10*A + .04*B + .13*C + .25*D + .36*E + .12*F$$

Result Retest Percentage

0.00% 0.00% 0.00%

Weight Inputs

Weight	Inputs
0.1	Test Plans Required
0.04	Test Procedures Required
0.13	Test Reports Required
0.25	Test Drivers Required
0.36	Integration Testing
0.12	Formal Testing

Least	Likely	Most	Percentage of the existing software that...
0%	0%	0%	... requires test plans to be rewritten
0%	0%	0%	... requires test procedures to be identified and written
0%	0%	0%	... requires documented test reports
0%	0%	0%	... requires test drivers and simulators to be rewritten
0%	0%	0%	... requires integration testing
0%	0%	0%	... requires formal demonstration testing

COTS / Open Source Component Integration With QuickSize

Inputs

Parameters

Function Based Sizing

Economic Factors

Project Monitor & Control Snapshots

Maintenance

Calibration

Labor Category Allocation

COTS:

Least

Likely

Most

Note

+

QUICK SIZE

-

OFF-THE-SHELF PRODUCT CHARACTERISTICS

Component Type

Plug-In Component

Component Volatility

Low

Nom

Hi

Component Application Complexity

Low

Low+

Nom

Interface Complexity

Low

Nom

Nom

Product Support

Hi

VHi

EHi

-

USE

Component Selection Completeness

Hi

Experience With Component

Low

Nom

Hi

Learning Rate

Nom

Hi

VHi

Reverse Engineering

VLo

Low

Low

Component Integrate and Test

VLo

VLo+

Low+

Test Level

Low+

Nom-

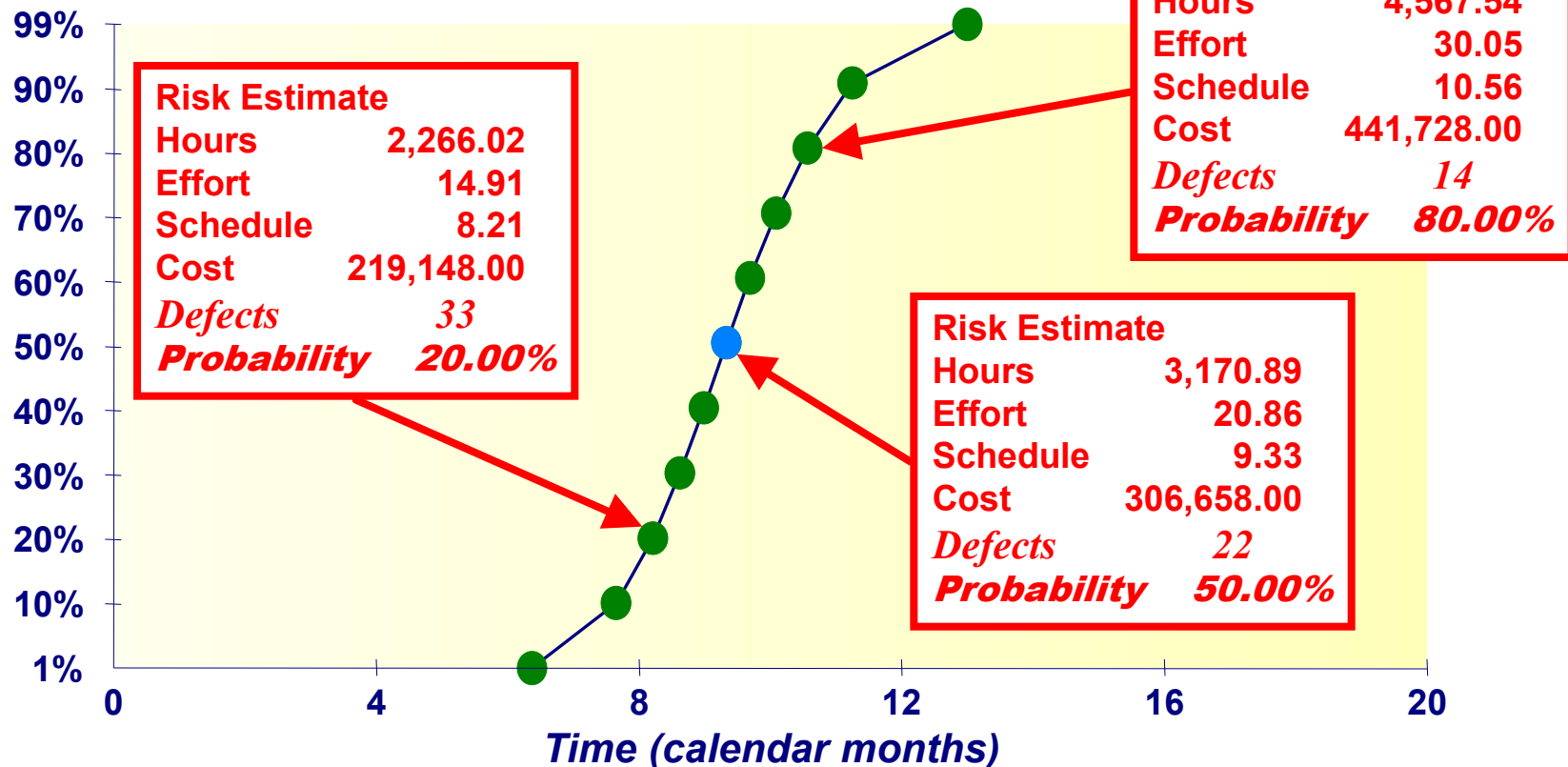
Nom

Risk Analysis Is Critical TO Understanding Full Modernization Costs

Schedule Probability

Example Application 1

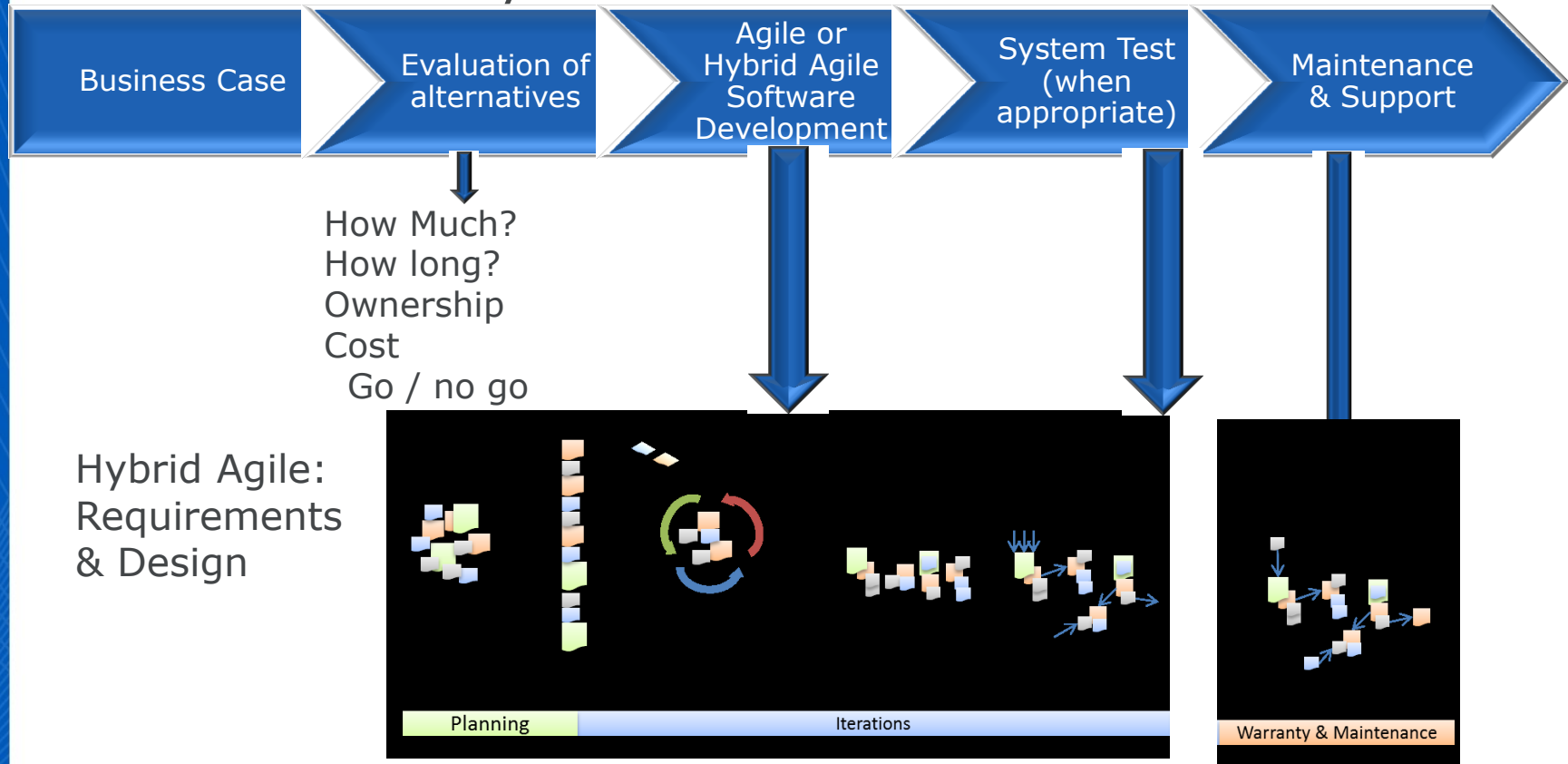
Probability



Agile Modernization Needs Estimates

#noestimates Viable For Detailed Development - Should Not Abdicate In Substantial Developments

For substantial systems



Agile development = root level software development management...

Story point estimating is short term productivity management

It is not a business decision making process

estimate

estimate • analyze • plan • control

Cloud Costing



We Know How To Estimate Cloud Costs and ROI



- Cloud isn't so different that alternate approaches to cost, ROI or business case are needed
- Important to identify costs that will increase as well as decrease.. E.g. bandwidth
- Risk must be factored in
 - E.g. data inaccessibility
- SaaS and on-premises setup costs could be similar..
 - No SaaS savings
- Measurement, estimation and ROI processes are essential to make the most viable decisions

When cloud computing is perceived as a panacea, with assumed savings, it's buyer beware
Jobs Changing, NOT Disappearing

On Premises Often Cheaper (IaaS Example)

In-house(Buy) \$8,873 total 5 years

Replacement Server: Dell PowerEdge T430 - **\$3,943**

Back-up Software License and agents (2 options)

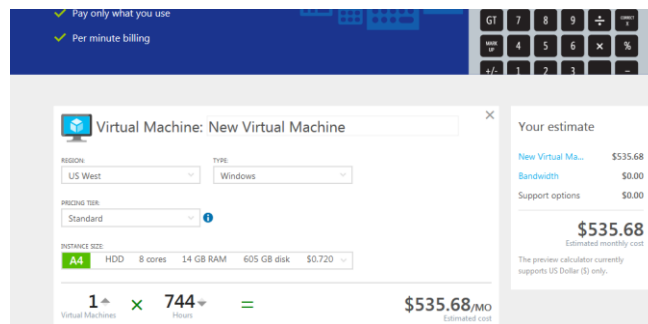
1. Symantec Back-up Exec: **\$2,822** (includes 2014 vr and 4 agents)

2. Dell NetVault - **\$2,108** (includes 1 TB capacity)

Note: Costs Here EXCLUDE IT Support Costs

\$8,873

Cloud (Rent) \$6,423 Annual



Virtual Machine: New Virtual Machine

REGION: US West TYPE: Windows

PRICING TIER: Standard

INSTANCE SIZE: A4 HDD 8 cores 14 GB RAM 605 GB disk \$0.720

1 x 744 = \$535.68/MO

Your estimate

New Virtual Ma...	\$535.68
Bandwidth	\$0.00
Support options	\$0.00
\$535.68	Estimated monthly cost

The preview calculator currently supports US Dollar (\$) only.

Your estimate

New Virtual Ma... \$535.68

Bandwidth

Support options

\$535.68

\$535.68 * 12

\$0.00

\$535.68

Estimated monthly cost

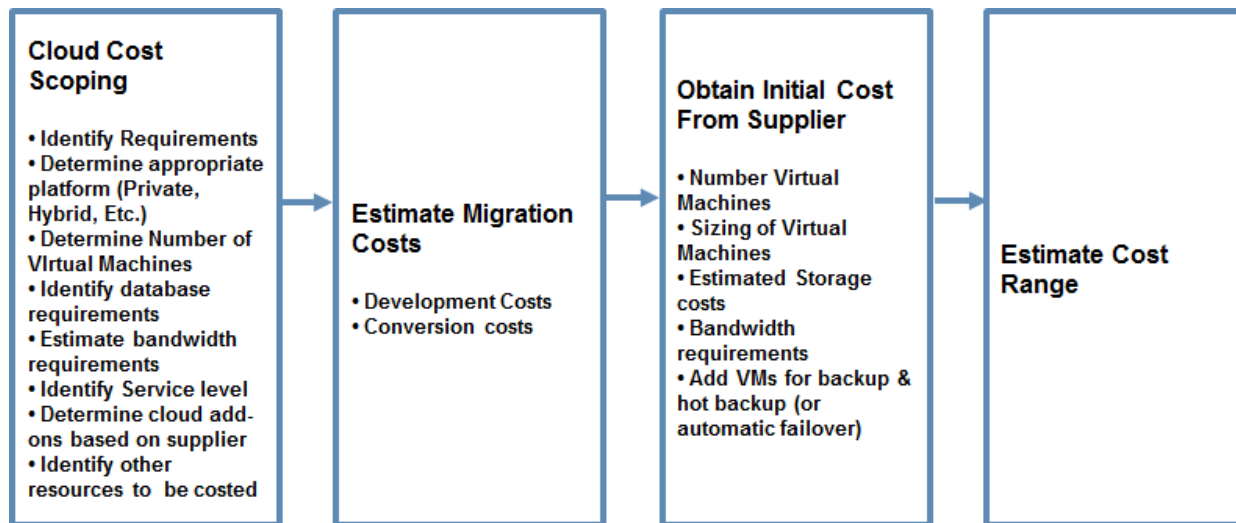
Note: Costs Here EXCLUDE IT Support Costs

Cloud Solutions Still Have Major Organizational Responsibilities & Costs

	IaaS	PaaS	SaaS
Corporate Data	Organization	Organization	Organization
Archival Backups	Organization	Organization	Organization
Local user support	Organization	Organization	Organization
Source Code	Organization	Organization	Vendor
Application Configuration	Organization	Organization	Maybe
Programming Languages	Organization	Vendor	Vendor
Frameworks	Organization	Vendor	Vendor
Containers	Organization	Vendor	Vendor
Operating System	Vendor	Vendor	Vendor
Hardware	Vendor	Vendor	Vendor
Service level agreements	Difficult or impossible	Difficult or impossible	Difficult or impossible

Note The Line between IaaS and PaaS is blurring to the point that is generally NOT a cost driver

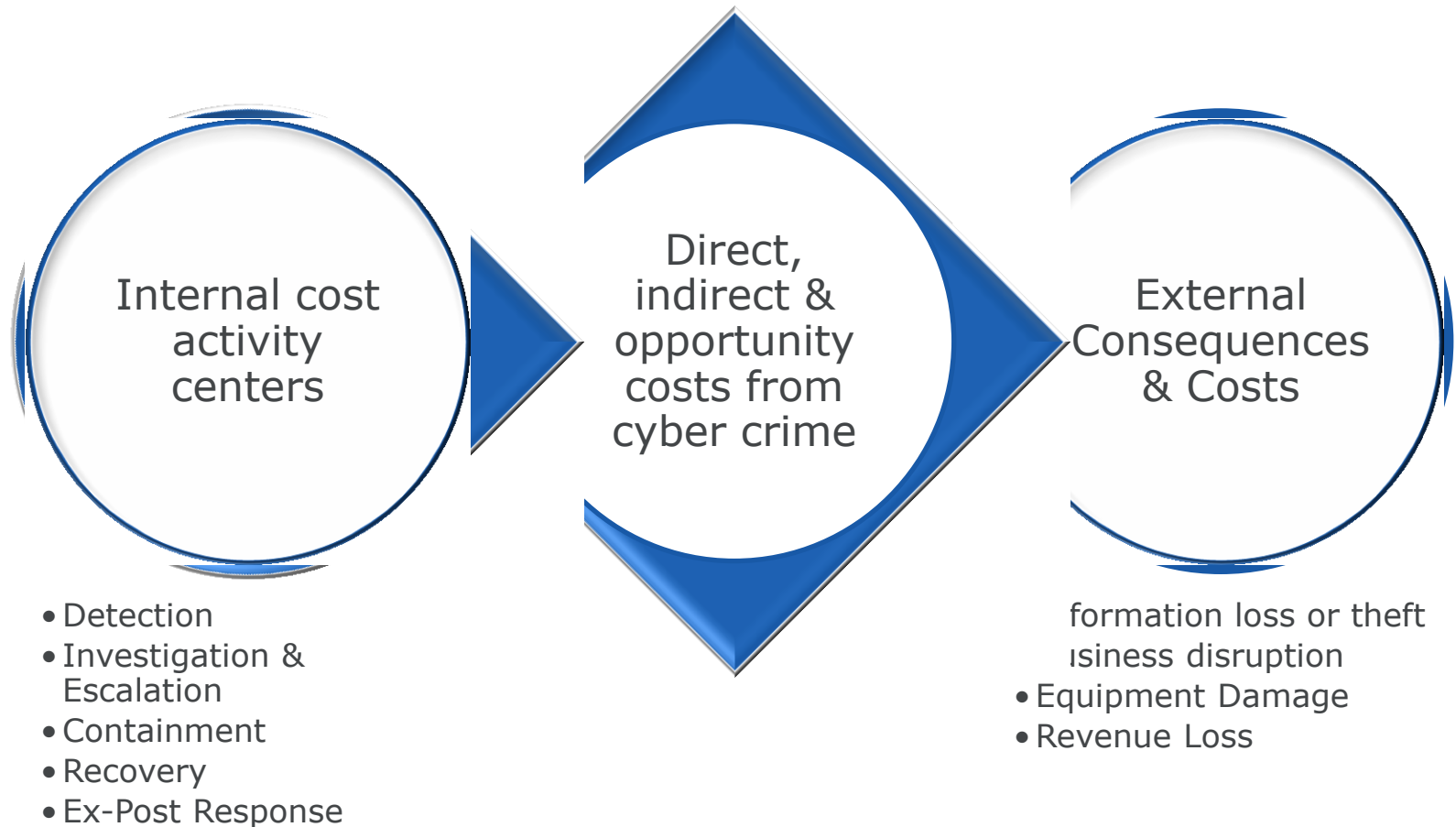
Cloud Costing Scope



Cloud Labor Costs

Cloud Labor Costs			
Cloud Does Not Relieve the Organization of Costs & Responsibilities			
	Integration as a Service (IaaS)	Platform as a Service (PaaS)	Software as a Service (SaaS)
Migration (One-Time) Costs	Possibly Development Data conversion Training	Development in new platform Data conversion Training	Data conversion Configuration Training
Corporate Data	Organization	Organization	Organization
Archival Backups (some new cloud tech may mitigate this)	Organization	Organization	Organization
Local user support	Organization	Organization	Organization
Source Code	Organization	Organization	Vendor
Application Configuration	Organization	Organization	Either
Programming Languages	Organization	Vendor	Vendor
Frameworks	Organization	Vendor	Vendor
Containers	Organization	Vendor	Vendor
Operating System	Vendor (Usually)	Vendor	Vendor
Hardware	Vendor	Vendor	Vendor

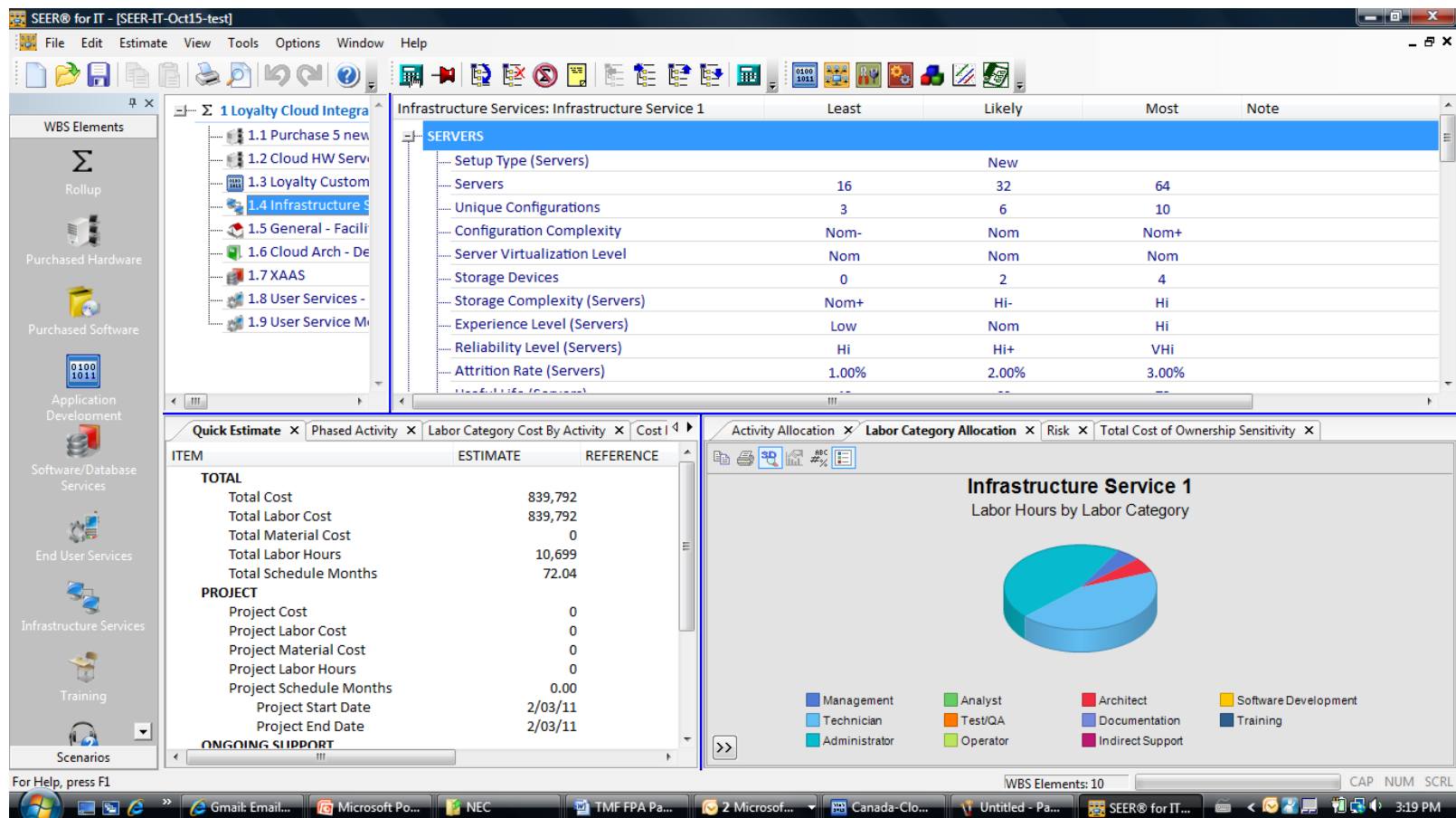
Direct, indirect & opportunity costs from cyber crime (Source HP)



SEER Modeling Can Estimate Total Ownership Costs For Software & IT



Size, approach, help desk, training, and many more IT / software cost modeling functions



Σ 1 Web Application Infrastructure

Σ 1.1 Multi-Tier Web Application

Σ 1.1.1 Web Tier

1.1.1.1 Web Server

1.1.1.2 Web Service

Σ 1.1.2 Application Tier

1.1.2.1 Application Host

1.1.2.2 Application

Σ 1.1.3 Data Tier

1.1.3.1 Database and Storage Host

1.1.3.2 Database Configuration

Σ 1.1.4 Procurement of Hardware and Software

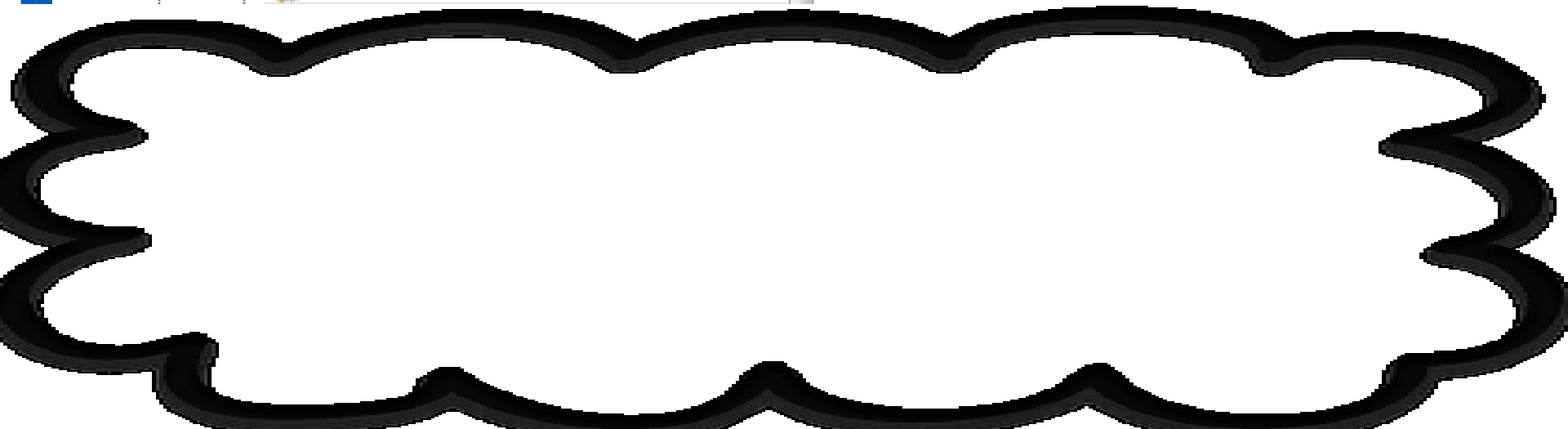
1.1.4.1 Production Servers Purchase

1.1.4.2 Purchase QA and Dev Server

1.1.4.3 SQL Server

WBS Total Ownership Cost

ELEMENT	PROJECT COST	ONGOING SUPP... COST	TOTAL COST
1 Web Application Infrastructure	481,762	4,859,396	5,341,158
1.1 Multi-Tier Web Application	390,446	4,785,606	5,176,052
1.2 Cloud Hosted Multi-Tier			
1.3 QA and Dev Environment	55,125	0	55,125
1.4 Training and Documentation	36,190	2,224	38,414
1.5 Support	0	71,567	71,567



Some Gottchas in Cloud Costing



- Reliability requirements can double cloud resources needed
- Security
- Hot backup can double cloud resources
- Is backup in cloud sufficient
- Will timing work with application being modernized

Key Points

Modernization
can be risky &
substantially
misestimated



Modernization
approaches can
reduce cost &
risk



Open source,
Agile, Cloud &
other technologies
can help... But
they come at a cost
(not free)



Project Success Limitations Reduce Productivity (Source Lawhorn, Project Failures)



- **Poorly defined applications** (miscommunication between business and IT) contribute to a 66% project failure rate, costing U.S. businesses at least \$30 billion every year (Forrester Research)
- **60% – 80% of project failures** can be attributed directly to poor requirements gathering, analysis, and management (Meta Group)
- **50%** are rolled back out of production (Gartner)
- **40%** of problems are found by end users (Gartner)
- **25% – 40%** of all spending on projects is wasted as a result of re-work (Carnegie Mellon)
- Up to **80%** of budgets are consumed fixing self-inflicted problems (Dynamic Markets Limited Study)

Technical Debt Is Exponential When Shipping Early

Defects Analysis - Program: Data Analyzer

Time Phased Defects

Months From Estimate	Delivery Date	Hours	Est. Cost	Delivered Defects			
-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	9/30/08	36,938	4,155,528	167			
-4	10/30/08	39,930	4,492,138	140			
-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		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			
3	5/30/09	60,938	6,855,538	34			
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