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

Step 1. Procure Key Performance Parameters that are inviolate
Step 2. Identify Affordability Goals & Weighted Figures of Merit
Step 3. Gather Requirements, Features, Performance
Step 4. Define Technical Baseline Alternatives & Assumptions
Step 5. Perform Technical Design Analysis for Each Alternative
Step 6. Perform Cost Schedule Analysis of Each Alternative
Step 7. Assess Benefits Based on Figures of Merit
Step 8. Perform Probabilistic Risk Analysis
Step 9. Assess Alternatives & Select Optimal Alternative
Step 10. Document Analysis and Lessons Learned

Pricing strategies assumed in step 7. Since price is a figure of merit
Step 1 Key Performance Parameters (KPPs)

• **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?

© 2013 Copyright Galorath Incorporated
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

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.

Reference Operations Concept

Initial Mass in LEO

Total Architecture Mass (kg/1000)
- Lander EDS #1
- Lander EDS #2
- Lander EDS #3
- Lander EDS #4
- Kick Stage
- Descent Stg
- Accent Stg
- CEV EDS #1
- CEV EDS #2
- CEV SM
- CEV CM
Cloud Example: But When We Look at Figures of Merit

- Is the cloud secure enough?
- Is the cloud fast enough?
- Is cloud vendor 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

- **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 Techniques

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

---

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

1. Identify assumptions
2. Rank order assumptions based on estimate impact
3. Identify high ranking assumptions that are risky
4. Clarify high ranking, high risk assumptions & quantify what happens if those assumptions change
5. Adjust range of SEER inputs to describe the uncertainty in assumption
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

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

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
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

- 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

1. Establish Estimate Scope
2. Establish Technical Baseline, Ground Rules, Assumptions
4. Refine Technical Baseline Into Estimable Components
4. Collect data / estimation inputs
5. Estimate Baseline Cost, Schedule, Affordability Value
6. Validate Business Case Costs & Benefits (go / no go)
6. Quantify Risks and Risk Analysis
8. Generate a Project Plan
10. Track Project Throughout Development
Basic Cost Estimating Process (Source CEBOK)

- **Work Breakdown Structure (WBS) Development**
- **Program/System Baseline Development**
US GAO process for Credible Estimates

Initiation and research
Your audience, what you are estimating, and why you are estimating it are of the utmost importance.

Assessment
Cost assessment steps are iterative and can be accomplished in varying order or concurrently.

Analysis
The confidence in the point or range of the estimate is crucial to the decision maker.

Presentation
Documentation and presentation make or break a cost estimating decision outcome.

Analysis, presentation, and updating the estimate steps can lead to repeating previous assessment steps.

1. Define the estimate's purpose
2. Develop the estimating plan
3. Define the program
4. Determine the estimating structure
5. Identify ground rules and assumptions
6. Obtain the data
7. Develop the point estimate and compare it to an independent cost estimate
8. Conduct sensitivity
9. Conduct a risk and uncertainty analysis
10. Document the estimate
11. Present estimate to management for approval
12. Update the estimate to reflect actual costs/changes

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

Concept
- Describe Idea & Possible benefits

Great Idea

Gate 1
Opportunity Analysis
- Determine customer acceptance
- Interviews focus groups, etc.

Gate 2
Marketing Analysis

Gate 3
Feasibility Study
- Design solution
- Estimate cost / schedule
- Analyze risk
- Determine feasibility / ROI

Gate 4
Committed Business Case
- Pilot or proof of concept
- Validate & commit to design & approach
  - Revised estimates & schedule
- Risk reduction
- Baselined plan

Achieve Business Case

Full Execution or Deployment
- Build solution
  - Deploy
- Achieve business case
- Capture lessons learned
  - incl: estimating

Risk

Investment in time and money
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

<table>
<thead>
<tr>
<th>Model Category</th>
<th>Description</th>
<th>Advantages</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guessing</td>
<td>Off the cuff estimates</td>
<td>Quick, Can obtain any answer desired</td>
<td>No Basis or substantiation, No Process, Usually Wrong</td>
</tr>
<tr>
<td>Analogy</td>
<td>Compare project with past similar projects.</td>
<td>Estimates are based on actual experience.</td>
<td>Truly similar projects must exist</td>
</tr>
<tr>
<td>Expert Judgment</td>
<td>Consult with one or more experts.</td>
<td>Little or no historical data is needed; good for new or unique projects.</td>
<td>Experts tend to be biased; knowledge level is sometimes questionable; may not be consistent.</td>
</tr>
<tr>
<td>Top Down Estimation</td>
<td>A hierarchical decomposition of the system into progressively smaller components is used to estimate the size of a software component.</td>
<td>Provides an estimate linked to requirements and allows common libraries to size lower level components.</td>
<td>Need valid requirements. Difficult to track architecture; engineering bias may lead to underestimation.</td>
</tr>
<tr>
<td>Model Category</td>
<td>Description</td>
<td>Advantages</td>
<td>Limitations</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Bottoms Up Estimation</td>
<td>Divide the problem into the lowest items. Estimate each item... sum the parts.</td>
<td>Complete WBS can be verified.</td>
<td>The whole is generally bigger than the sum of the parts. Costs occur in items that are not considered in the WBS.</td>
</tr>
<tr>
<td>Design To Cost</td>
<td>Uses expert judgment to determine how much functionality can be provided for given budget.</td>
<td>Easy to get under stakeholder number.</td>
<td>Little or no engineering basis.</td>
</tr>
<tr>
<td>Simple CER’s</td>
<td>Equation with one or more unknowns that provides cost / schedule estimate.</td>
<td>Some basis in data.</td>
<td>Simple relationships may not tell the whole story. Historical data may not tell the whole story.</td>
</tr>
<tr>
<td>Comprehensive Parametric Models</td>
<td>Perform overall estimate using design parameters and mathematical algorithms.</td>
<td>Models are usually fast and easy to use, and useful early in a program; they are also objective and repeatable.</td>
<td>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.</td>
</tr>
</tbody>
</table>
Affordability Alternatives Generally Provide ROM Estimates (Source APMP: Just Say No)

Phase -1 0 1 2 3 4

Phase 0

ROM

ROM

ROM

ROM

ROM

EARLY ESTIMATING
3-5 people, 3 - 5 days
Top Down, parametric model based price estimating
Vs.
Current state: 90 people, 6wks

Formal Bid Gate 4
15-20 people
4 weeks
(Bid Stds + History)

Modified Budgetary Estimate
Draft RFP/Gate 3
6-8 people, 3 weeks
(Bid Stds + History)

Draft RFP
RFP
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
Five Specific Areas of Concern:

- **Target Affordability and Control Cost Growth**
- **Reduce Non-Productive Processes and Bureaucracy**
- **Incentivize Productivity and Innovation in Industry**
- **Promote Real Competition**
- **Improve Tradecraft in Services Acquisition**
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”

http://hbr.org/2011/09/why-your-it-project-may-be-riskier-than-you-think/ar/1
Step 7. Assess Benefits Based on Figures of Merit

- Return on Investment often main criterion in IT systems
## ROI Analysis of A New System

**Cost of capital**: 8.0%

<table>
<thead>
<tr>
<th></th>
<th>Initial Investment</th>
<th>Year 1</th>
<th>Year 2</th>
<th>Year 3</th>
<th>Year 4</th>
<th>Year 5</th>
<th>Year 6</th>
<th>Year 7</th>
<th>Total Ownership</th>
</tr>
</thead>
<tbody>
<tr>
<td>Investment</td>
<td>$100,000</td>
<td>($40,000)</td>
<td>$60,000</td>
<td>$110,000</td>
<td>$100,000</td>
<td>$100,000</td>
<td>$150,000</td>
<td>$150,000</td>
<td>$630,000</td>
</tr>
<tr>
<td>Increase/(dec.) in revenue</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Increase/(dec.) in op. exp.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cash Flow</td>
<td>($100,000)</td>
<td>($130,000)</td>
<td>($10,000)</td>
<td>$40,000</td>
<td>$78,000</td>
<td>$76,000</td>
<td>$123,000</td>
<td>$122,000</td>
<td>$199,000</td>
</tr>
<tr>
<td>PV of Cash Flow</td>
<td>($100,000)</td>
<td>($120,370)</td>
<td>($8,573)</td>
<td>$31,753</td>
<td>$57,332</td>
<td>$51,724</td>
<td>$77,511</td>
<td>$71,186</td>
<td>$60,563</td>
</tr>
<tr>
<td>NPV</td>
<td>60,563</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRR</td>
<td>13.5%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROI</td>
<td>121%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A Complete ROI analysis should analyze risk and uncertainty as well as likely
Technical Debt (Source: CAST)

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)

“Man will never reach the moon regardless of all future scientific advances.”
- Dr. Lee DeForest, Inventor of Television

“Airplanes are interesting toys but of no military value.”
- Marechal Ferdinand Foch, Professor of Strategy, Ecole Superieure de Guerre

“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 Schwarzkoph, 1991

“There is no reason anyone would want a computer in their home.”
- Ken Olson, president and founder of Digital, 1977

“640K ought to be enough for anybody.”
- Bill Gates, 1981

“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”

Objective (Goal) vs. Threshold (No Greater Than)

- Region for Marginal Performance Improvement
- Region for “Best Bang for Buck”
- High Cost Payoff Small Performance Penalty... Consider

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
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
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

Statistician Drowns in River with Average Depth of 3 Feet!
Agile Uncertainty May Be The Same or Worse With Agile

• Precision comes over time! And what that it is unclear
Range vs. Point Estimates
(Source US Army)

Range of Risk & Uncertainty

Range estimate provides a degree of risk and uncertainty

Point estimate is most likely within range estimate with higher potential for cost increase

ROM -30% to +75%
Analogy -15% to +30%
Parametric -10% to +20%
Engineering -5% to +15%
Actual -3% to +10%
Target Cost

Technical and Program Maturity

Estimating Accuracy

Materiel Solution Analysis
Technology Development
Engineering and Manufacturing Development
Production & Deployment
Operations & Support

Pre-Systems Acquisition
Systems Acquisition
Sustainment
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”
Communications Are Challenging and Get Worse as Number of Organizations & Staff Increase

\[ \frac{n(n - 1)}{2} \]

- 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

Example early ship shows 400%+ more defects than recommended.

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

---

**Defects Analysis - Program: Data Analyzer**

<table>
<thead>
<tr>
<th>Months From Estimate</th>
<th>Delivery Date</th>
<th>Hours</th>
<th>Est. Cost</th>
<th>Delivered Defects</th>
<th>Defect Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>-8</td>
<td>6/30/08</td>
<td>28,330</td>
<td>3,187,117</td>
<td>268</td>
<td></td>
</tr>
<tr>
<td>-7</td>
<td>7/30/08</td>
<td>31,412</td>
<td>3,501,165</td>
<td>230</td>
<td></td>
</tr>
<tr>
<td>-6</td>
<td>8/30/08</td>
<td>33,996</td>
<td>3,824,578</td>
<td>197</td>
<td></td>
</tr>
<tr>
<td>-5</td>
<td>9/30/08</td>
<td>36,938</td>
<td>4,155,528</td>
<td>167</td>
<td></td>
</tr>
<tr>
<td>-4</td>
<td>10/30/08</td>
<td>39,930</td>
<td>4,492,138</td>
<td>140</td>
<td></td>
</tr>
<tr>
<td>-3</td>
<td>11/30/08</td>
<td>42,956</td>
<td>4,832,523</td>
<td>117</td>
<td></td>
</tr>
<tr>
<td>-2</td>
<td>12/30/08</td>
<td>45,996</td>
<td>5,174,829</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>1/30/09</td>
<td>49,042</td>
<td>5,517,264</td>
<td>85</td>
<td></td>
</tr>
<tr>
<td>Estimate</td>
<td>3/02/09</td>
<td>52,061</td>
<td>5,856,845</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3/30/09</td>
<td>55,073</td>
<td>6,195,760</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4/30/09</td>
<td>58,033</td>
<td>6,528,697</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5/30/09</td>
<td>60,938</td>
<td>6,855,538</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6/30/09</td>
<td>63,778</td>
<td>7,175,022</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>7/30/09</td>
<td>66,542</td>
<td>7,486,020</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>8/30/09</td>
<td>69,223</td>
<td>7,787,520</td>
<td>16</td>
<td></td>
</tr>
</tbody>
</table>
Software Progress and VIABLE SHIP DATE Can Be Determined By Defect Insertion & Removal

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

Increased defect reporting rate shows a worsening trend.

Track defect discovery and removal rates against expected rates.
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

1. Establish Estimate Scope
2. Establish Technical Baseline, Ground Rules, Assumptions
3. Collect Data
4. Estimate and Validate Software Size
5. Prepare Baseline Estimates
6. Review, Verify and Validate Estimate
7. Quantify Risks and Risk Analysis
8. Generate a Project Plan
9. Document Estimate and Lessons Learned
10. Track Project Throughout Development

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

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

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
  - \[ P(\text{N elements Successful}) = (A_{\text{prob}})(B_{\text{prob}})\ldots(N_{\text{prob}}) \]
  - For just 3 independent elements each with a 90% probability
    - \[ P(\text{3 Elements Successful}) = (.9)(.9)(.9) = .729 \]
- 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)

<table>
<thead>
<tr>
<th>Process</th>
<th>Durations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step</td>
<td>Expected</td>
</tr>
<tr>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>90</td>
</tr>
<tr>
<td>6</td>
<td>25</td>
</tr>
<tr>
<td>7</td>
<td>35</td>
</tr>
<tr>
<td>8</td>
<td>45</td>
</tr>
<tr>
<td>9</td>
<td>70</td>
</tr>
<tr>
<td>10</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>500</td>
</tr>
</tbody>
</table>

What would you forecast the schedule duration to be?
Range Clarifies Risk -2 (Adapted from SEI)

<table>
<thead>
<tr>
<th>Process</th>
<th>Durations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best</td>
</tr>
<tr>
<td>Step</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Risk Analysis Makes Projects More Successful – 3
(Adapted from SEI)

50% confidence, the project will be under 731 days duration

With 90% confidence, the project will be under 817 days duration

ORIGINAL: Almost guaranteed to exceed the 500 day duration.
Managing Risk Improves Results


Source: PA Consulting
Survey of Global Banks
Step 9 Assess Alternatives & Select

- 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)

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Importance Weight (%)</th>
<th>Rating</th>
<th>Weighted Rating</th>
<th>Rating</th>
<th>Weighted Rating</th>
<th>Rating</th>
<th>Weighted Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>high efficiency</td>
<td>30</td>
<td>4</td>
<td>1.20</td>
<td>2</td>
<td>0.60</td>
<td>3</td>
<td>0.90</td>
</tr>
<tr>
<td>high reliability</td>
<td>25</td>
<td>4</td>
<td>1.00</td>
<td>3</td>
<td>0.75</td>
<td>3</td>
<td>0.75</td>
</tr>
<tr>
<td>low maintenance</td>
<td>20</td>
<td>4</td>
<td>0.80</td>
<td>3</td>
<td>0.60</td>
<td>2</td>
<td>0.40</td>
</tr>
<tr>
<td>low cost</td>
<td>15</td>
<td>2</td>
<td>0.30</td>
<td>4</td>
<td>0.60</td>
<td>3</td>
<td>0.45</td>
</tr>
<tr>
<td>light weight</td>
<td>10</td>
<td>2</td>
<td>0.20</td>
<td>4</td>
<td>0.40</td>
<td>3</td>
<td>0.30</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>NA</td>
<td>3.50</td>
<td>NA</td>
<td>2.95</td>
<td>NA</td>
<td>2.80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rating</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsatisfactory</td>
<td>0</td>
</tr>
<tr>
<td>Just tolerable</td>
<td>1</td>
</tr>
<tr>
<td>Adequate</td>
<td>2</td>
</tr>
<tr>
<td>Good</td>
<td>3</td>
</tr>
<tr>
<td>Very Good</td>
<td>4</td>
</tr>
</tbody>
</table>
Example: Traditional On Premises Software Total Ownership Cost Allocation

IT Services & Infrastructure Are Situational but Generally 60% of TOC

Development = Biggest Risk

- Software Development
- Software Maintenance
- IT Infrastructure
- IT Services

For Cloud Some Costs reduced or eliminated.. Other new Costs occur

© 2013 Copyright Galorath Incorporated
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**
- **Software Maintenance**
- **IT Infrastructure & Services**

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

Assume $300m development could be over $3b total ownership... But it must be done
Cloud Example: Current Costs of IaaS Are Readily Available

TCO Comparison Calculator for Web Applications (Beta)

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

<table>
<thead>
<tr>
<th></th>
<th>On-Premises</th>
<th>AWS</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Servers</td>
<td>$26,579</td>
<td>$13,547</td>
<td>$13,031</td>
</tr>
<tr>
<td>Storage</td>
<td>$47,919</td>
<td>$11,192</td>
<td>$36,728</td>
</tr>
<tr>
<td>Network</td>
<td>$25,767</td>
<td>$972</td>
<td>$24,795</td>
</tr>
<tr>
<td>Environment</td>
<td>$93,150</td>
<td>$0</td>
<td>$93,150</td>
</tr>
<tr>
<td>Administration</td>
<td>$60,720</td>
<td>$15,180</td>
<td>$45,540</td>
</tr>
<tr>
<td><strong>Total / year</strong></td>
<td><strong>$254,135</strong></td>
<td><strong>$40,891</strong></td>
<td><strong>$213,244</strong></td>
</tr>
</tbody>
</table>

Region: US East (Northern Virginia)
Usage Pattern: Spikey Predictable

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

© 2013 Copyright Galorath Incorporated
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

© 2013 Copyright Galorath Incorporated
Estimation Best Practices
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
Backup Slides