

# Workbook

## MEASURING RESULTS

### Is Consolidation Paying Off?

BY PAUL A. STRASSMANN

One way to tell whether your effort to consolidate software operations is effective: Compare the change in function points, which is a measure of a software application's complexity, as well as the change in software costs and number of personnel—before and after consolidation. The example below assumes a company with an I.T. budget of about \$150 million and involves the adoption of a service-oriented architecture, along with data warehouses and a portal. Here, personnel for software maintenance drops by 92%, while resources for new development go up 60%.

**INSTRUCTIONS:** Fill in your organization's numbers, describing how you expect your key indicators to change from the Base Year to the Projected Year. Then follow the calculations defining the transformation indicators. You can also download an interactive version of this worksheet, including an explanation of underlying assumptions, from our Premium Tools Library at [GO.BASELINEMAG.COM/MAR07](http://GO.BASELINEMAG.COM/MAR07).

#### Tool: Measuring Your Software Consolidation Strategy

	BASICS	EXAMPLE	YOUR FIRM
<b>BEFORE CONSOLIDATION</b>			
<b>BASE YEAR</b>		<b>2003</b>	
A	Number of enterprise applications (@ 100,000 function points each)	1	
B	Number of major applications (@ 15,000 function points each)	10	
C	Number of minor applications (@ 5,000 function points each)	5	
D	Number of maintenance applications (@ 100 function points each)	25	
E	Number of databases (@ 2,000 function points each)	15	
F	Number of networks (@ 15,000 function points each)	4	
G	Total number of function points $((A \times 100,000) + (B \times 15,000) + (C \times 5,000) + (D \times 100) + (E \times 2,000) + (F \times 15,000))$	367,500	
H	% of function points for software development (@ \$1,500 per function point)	5%	
I	% of functions points for maintenance (@ \$150 per function point)	95%	
J	Total software cost $((G \times H \times 1,500) + (G \times I \times 150))$	\$79,931,250	
K	Number of full-time equivalents for development; assumes fully loaded salary of \$125,000 $((G \times H \times 1,500) \div 125,000)$	221	
L	Number of full-time equivalents for maintenance; assumes fully loaded salary of \$75,000 $((G \times I \times 150) \div 75,000)$	698	
<b>AFTER CONSOLIDATION</b>			
<b>PROJECTED YEAR</b>		<b>2010</b>	
M	Number of major applications (@ 15,000 function points)	5	
N	Number of minor applications (@ 5,000 function points)	2	
O	Number of maintenance applications (@ 100 function points)	5	
P	Number of portal-driven applications (@ 5 function points)	50	
Q	Number of data warehouses (@ 75,000 function points)	1	
R	Number of networks (@ 75,000 function points)	1	
S	Total number of function points $((M \times 15,000) + (N \times 5,000) + (O \times 100) + (P \times 5) + (Q \times 75,000) + (R \times 75,000))$	235,750	
T	% of function points for development (@ \$750 per function point)	25%	
U	% of functions points for maintenance (@ \$25 per function point)	75%	
V	Total software cost $((S \times T \times 750) + (S \times U \times 25))$	\$48,623,438	
W	Number of full-time equivalents for development; assumes fully loaded salary of \$125,000 $((S \times T \times 750) \div 125,000)$	354	
X	Number of full-time equivalents for maintenance; assumes fully loaded salary of \$75,000 $((S \times U \times 25) \div 75,000)$	59	
<b>CHANGES RESULTING FROM CONSOLIDATION</b>			
Y	Difference in number of function points $-((G - S) \div G)$	-36%	
Z	Difference in total software cost $-((J - V) \div J)$	-39%	
AA	Difference in personnel devoted to software development $-((K - W) \div K)$	60%	
BB	Difference in personnel devoted to software maintenance $-((L - X) \div L)$	-92%	