Valuation of Software Intangible Assets

Eric A. Thornton
Senior Associate
(703) 917-6616
eathornton@willamette.com

ASA International Conference
San Diego, California
August 28, 2002

Valuation of Software Intangible Assets

Presentation Outline

1. Description of data processing intangible assets
2. Reasons to value computer software
3. Types of computer software
4. Typical valuation approaches
5. Remaining useful life
6. Special considerations
7. Illustrative example

Valuation of Software Intangible Assets

Data Processing Intangible Assets

Computer software
- For sale or license (product software)
- For internal use (operational software)

Automated databases

Intellectual Property
- Copyright protection for software and databases
- Copyright protection for multimedia works
- Protection of mask works under the Semiconductor Chip Protection Act of 1984
- Patent and trade secret protection for software
- Trademarks and trade names (packaged software, domain name)

Valuation of Software Intangible Assets

Reasons to Value Computer Software

1. Transactional reasons – establishing a purchase price, royalty rate, transfer price
2. Financing reasons – assessing collateral value, value as part of a solvency opinion
3. Taxation reasons – purchase price amortization, charitable contribution, assessment value for ad valorem property taxation
4. Bankruptcy reasons – assessing solvency of the software owner/user, identification of licensing/spin-off opportunities
5. Controversy reasons – quantifying copyright infringement/breach of contract damages
6. Management information reasons – identifying, quantifying, and managing the value of the intellectual property

Valuation of Software Intangible Assets

Types of Software

System Software

- Operating systems: Windows 95
- Languages: COBOL, C++
- Utilities: Anti-virus, backup

Business Operation Applications

- Accounting: Payroll, general ledger
- Manufacturing Control: Bill of materials, inventory control
- Engineering: Computer-aided design

Valuation of Software Intangible Assets

Types of Software (cont.)

Office Automation

- Word Processing: WordPerfect
- Spreadsheet: Excel
- Groupware: Lotus Notes

Educational/Recreational

- Reference: Encyclopedia, atlas
- Tutorials: Foreign language, math
- Games: Card/board games, simulators

Software Valuation Approaches/Methods

Cost Approach

- Reproduction cost vs. replacement cost
- Trended historical cost method
- Software engineering model method

Income Approach

- Income or cost savings
- Discounted cash flow method
- Relief from royalty method

Market Approach

- Availability & comparability issues
- Market transaction method
- Market replacement cost method

Software Engineering Models

COCOMO

- Constructive Cost Model
- Barry W. Boehm, *Software Engineering Economics*, 1981
- COCOMO II – 1997, re-calibrated annually, Center for Software Engineering, University of Southern California

SLIM

- Software Lifecycle Management
- Lawrence H. Putnam, Quantitative Software Management, Inc. (QSM)
- Computerized model, for license

Checkpoint/KnowledgePLAN

- Capers Jones, Software Productivity Research, Inc. (SPR)
- Computerized models, for license

Typical SW Engineering Model Inputs

Metrics
- Lines of code
- Function points (IFPUG, Mk II)
- Feature points (SPR)

Platform/Environment
- Hardware, operating system
- Language, tools, utilities
- Development practices

Product
- Type of application
- Complexity, required reliability
- Lifecycle (debugging, documentation)

Personnel
- Capability and experience
- Fully-loaded cost

Valuation of Software Intangible Assets

Income Approach

Discounted Cash Flow Method

- Need identifiable income stream
- Usually used for product software or databases that are sold or licensed
- Project income, expenses (excluding amortization/depreciation), and capital expenditures over remaining economic life (may differ from tax life)
- Include capital charge when additional assets are used to produce the revenues
- Discount to a present value

Relief from Royalty Method

- Similar to DCF but project cost savings to the owner, who would otherwise have to pay royalties on revenues earned from the asset
- Empirical analysis of market-derived royalty rates for comparable assets

Valuation of Software Intangible Assets

Market Approach

Market Transaction Method

- Arm's-length market transaction data
- Typically expressed as dollars per line of code or dollars per function point
- Apply to subject software lines of code or function points to estimate value
- Adjust for material differences between comparable and subject software
- Due to lack of data available, often used only as reasonableness check

Market Replacement Cost Method

- Contemplates replacement cost of the software in the open market
- Cost of comparable commercial off-the-shelf software packages
- Request hypothetical proposals to develop comparable software
- Consider the value of proprietary rights

Valuation of Software Intangible Assets

Obsolescence

When an intangible asset is less useful than its ideal replacement, its value should be adjusted to reflect loss of economic value due to:

- Functional obsolescence: impairment of its functional utility according to current market standards
- Technological obsolescence: inefficient or outdated programming language or hardware
- Economic obsolescence: market conditions, competitors
- Physical deterioration: not generally applicable to data processing intangibles

Valuation of Software Intangible Assets

Remaining Useful Life Analysis

- Income approach – to estimate the projection period
- Cost approach – to estimate the amount of obsolescence, if any
- Market approach – to select, reject, or make adjustments to comparable/guideline transactional data

Factors affecting expected remaining useful life of data processing intangible assets:

- Legal determinants
- Contractual determinants
- Functional determinants
- Technological determinants
- Economic determinants
- Analytical determinants

Valuation of Software Intangible Assets

Typical Remaining Life Analysis Method

- Theory developed at Iowa State University in the early 1900s
- Estimate historical attrition rate
- Develop survivor curves based on historical attrition rates
- Match subject survivor curves to Iowa (or other) curves, using least squares method
- Resulting curve and average life are used to predict future mortality
- Similar to actuarial analysis performed by insurance companies to estimate human life spans

For data processing intangibles:

- Data are not often available for this type of analysis
- Could examine "retirement" of lines of code (rewritten or replaced), objects, modules, programs, or subsystems

Valuation of Software Intangible Assets

Special Considerations

Ownership

- Source code for software packages is frequently sold/licensed to companies that modify the programs
- Do not include lines of code not owned in software engineering models

Obsolete and Duplicate Code

- Remove obsolete code from analysis
- Do not count multiple copies/versions of same program (adjust for "cloning")

Software Development Life Cycle

- Consider which phases of the life cycle have been completed and which are included in the model you are using

Software Metric Counting Conventions

- Use counting conventions consistent with the engineering model you use

Illustrative Example – COCOMO Inputs

1. Delivered Source Instructions – generally, source program lines of code excluding comments

2. Development Mode
   - Organic (e.g., business models)
   - Semidetached – (e.g., inventory)
   - Embedded – (e.g., avionics)

3. Cost Drivers
   - Product attributes
   - Computer attributes
   - Personnel attributes
   - Project attributes

Valuation of Software Intangible Assets

COCOMO Effort Equations

Organic: $MM = 3.2\ (KDSI)^{1.05} \times EAF$

Semidetached: $MM = 3.0\ (KDSI)^{1.12} \times EAF$

Embedded: $MM = 2.8\ (KDSI)^{1.20} \times EAF$

Where:

MM = Man-months

KDSI = Thousands of delivered source instructions

EAF = Effort adjustment factor (product of the cost driver effort multipliers)

Valuation of Software Intangible Assets

COCOMO Cost Driver Examples

Programming Language Experience:

Rating	Description	Multiplier
Very low	< 1 month	1.14
Low	4 months	1.07
Nominal	1 year	1.00
High	3 years	0.95

Required Software Reliability:

Rating	Description	Multiplier
Very low	Slight inconvenience	0.75
Low	Low, easily recoverable losses	0.88
Nominal	Moderate, recoverable losses	1.00
High	High financial loss	1.15
Very high	Risk to human life	1.40

Valuation of Software Intangible Assets

COCOMO – Cost to Develop

Application	DSI	MM Organic	MM Semi.
Accts payable	29,104		42
Accts receivable	35,982		53
Bank Rec.			
Fixed assets	23,617	28	
General ledger	32,753		48
Totals	121,456	28	143
Documentation Adj. (9%)		(3)	(13)
		25	130
Total man-months			155
Cost per man-month			$8,750
Cost to develop			$1,356,250

- EAF = 0.32
- $60,000 average annual salary
- 75% benefits and overhead

Valuation of Software Intangible Assets

SLIM Inputs

1. Effective source lines of code
2. Application type(s): business, scientific, telecommunications, system software, etc.
3. Customization of the life cycle
   - 4 phases: Feas, FC, MB, Maint
   - 10 milestones
4. Options: reliability, accounting, and documentation
5. Productivity adjustments: Tools/methods, technical difficulty, personnel, reuse
6. Constraints
   - Maximum time, effort, cost, peak staff
   - Minimum peak staff, mean time to defect

Valuation of Software Intangible Assets

Synthesis of SLIM and COCOMO Cost Approach Values

Application	PM SLIM	MM COCOMO
Accts payable	38.10	42
Accts receivable	51.73	53
Fixed assets	26.55	28
General ledger	45.40	48
Totals	161.78	171
Documentation adj.	(14.56)	(16)
	147.22	155
Cost per PM/MM	$ 8,750	$ 8,750
Cost to develop	$1,288,175	$1,356,250

Indicated Value Using Equal Weighting:

$1,322,213

$1,320,000 (rounded)