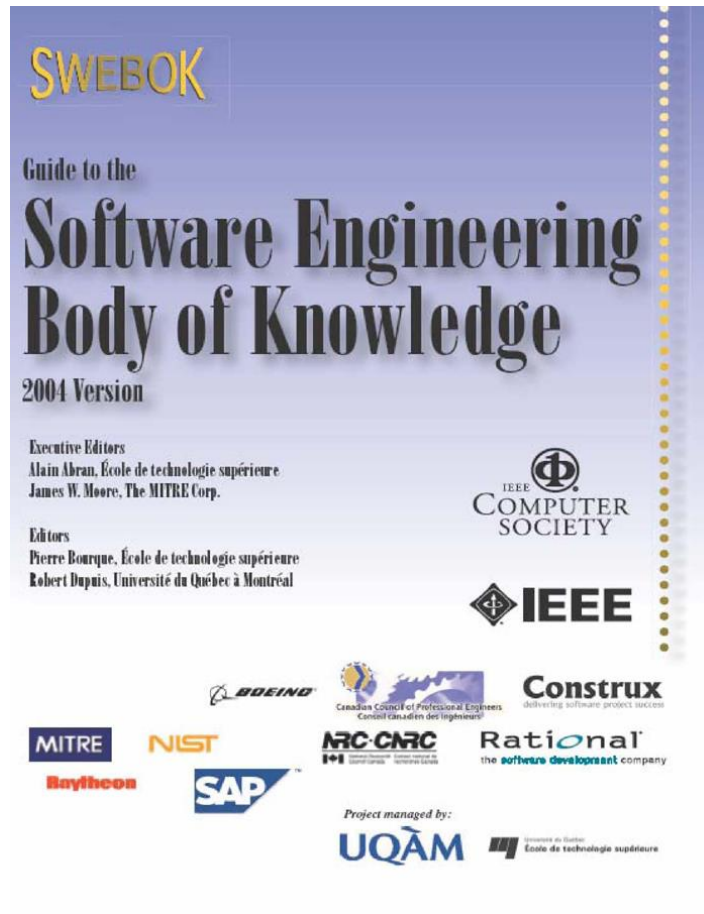


SWEBOK Guide V3 & Software Engineering Economics



SWEBOK Guide



15 Knowledge Areas in V3

- Requirements
- Design
- Construction
- Testing
- Maintenance
- Configuration management
- Engineering management
- Engineering process
- **Methods**
- **Quality**
- **Professional practice**
- **Engineering economics**
- **Computing foundations**
- **Mathematical foundations**
- **Engineering foundations**

SWEBOK Guide V3: Changes

- 10 years of industry advancement
 - Recognize Agile development
 - New methods and tools
 - ...
- Increase emphasis on software security
- Align with CSDA/CSDP knowledge areas
- Consolidated List of References

Certified Software Development Associate (CSDA)



- Entry level certification
 - No prerequisites
 - Equivalent of 4-year Software Engineering undergraduate degree
- No recertification needed

Certified Software Development Professional (CSDP)



- Mid-career level certification
 - **Education** requirement
 - Bachelor's degree
 - CSDA
 - Post-baccalaureate-level teaching experience, or
 - IEEE full membership
 - **Experience** requirement
 - Advanced degree in software engineering and at least two years experience, or
 - At least four years experience
- **Recertification** every 3 years

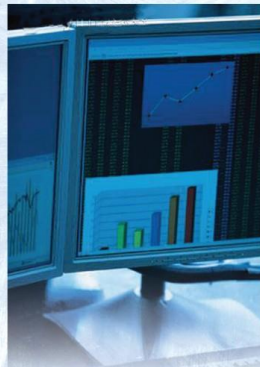
Steve Tockey

Foreword by Meilir Page-Jones

"This pioneering book highlights critical, overlooked skills needed by true software professionals."

—Steve McConnell, CEO and
Chief Software Engineer,
Construx Software

Return on Software



*Maximizing
the Return
on Your Software
Investment*

Business on Purpose

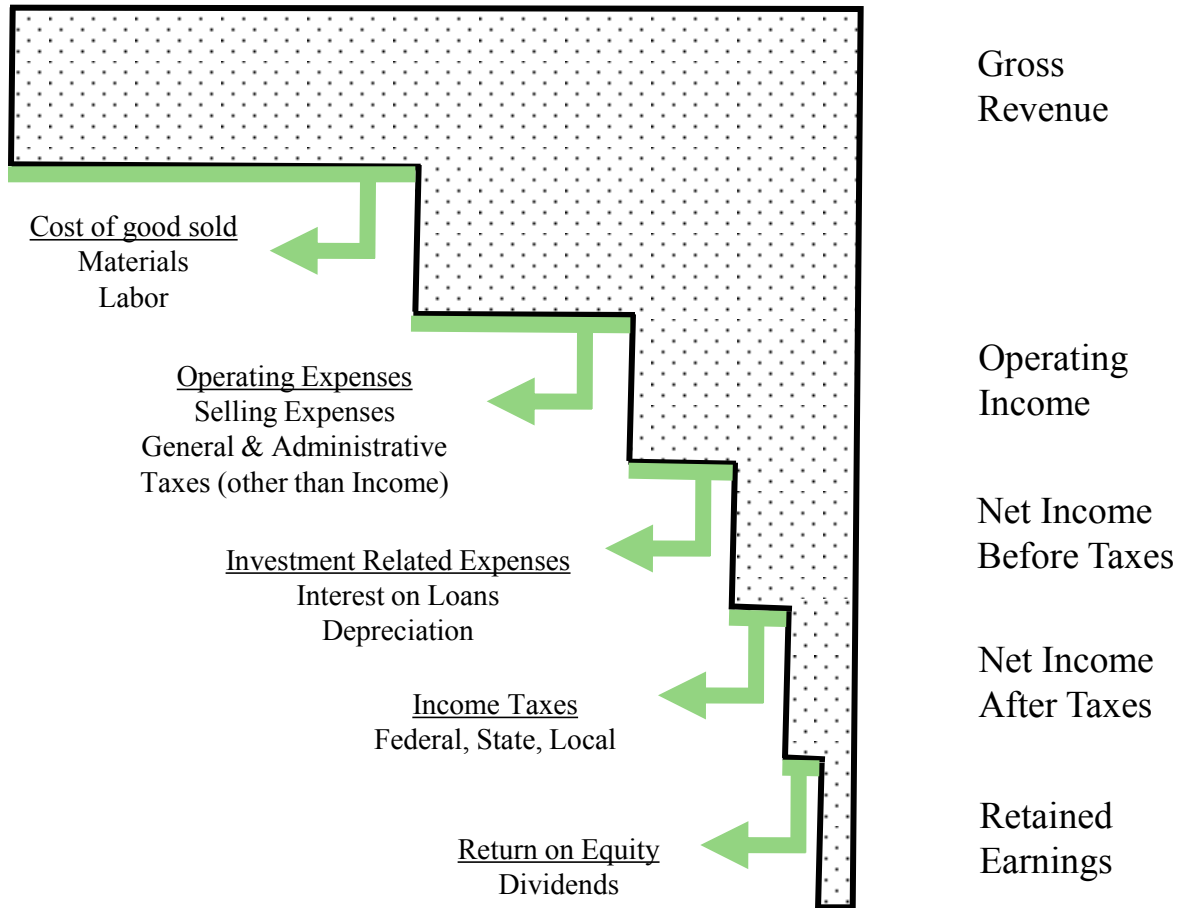
Why are companies in business?

Because it's fun? Educational? A way to have a positive impact on society?

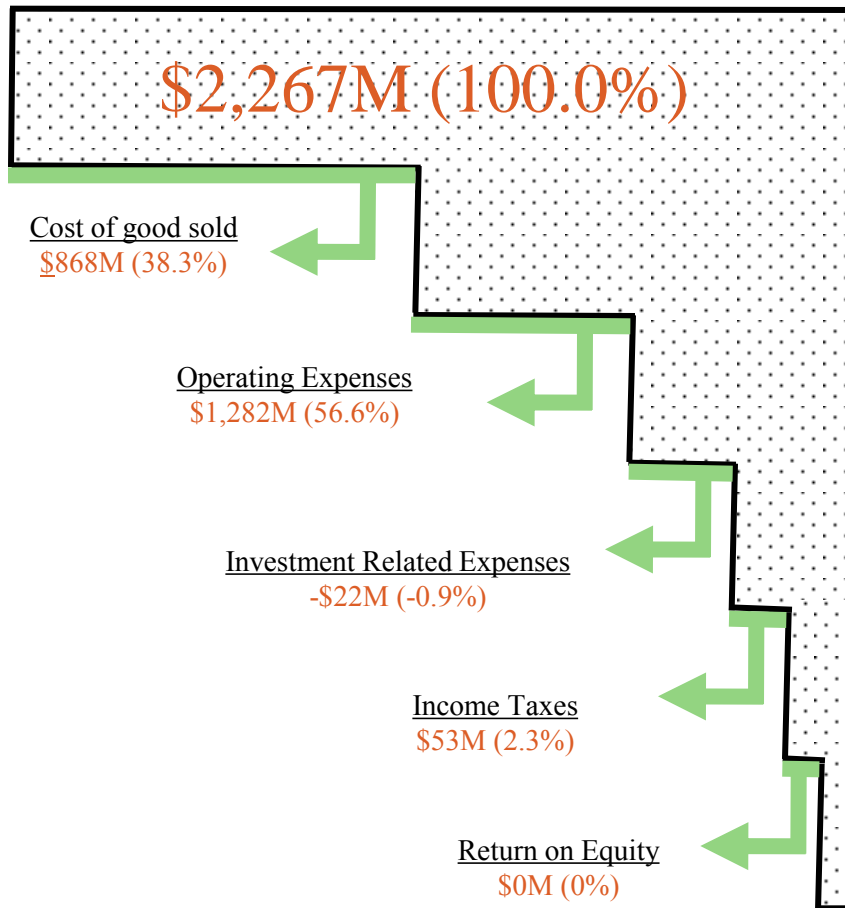
Companies are in business to **make a profit** for the owners

Where does the money come from?

Where Does the Money Go?



Peoplesoft, 2003



Gross
Revenue

Operating
Income \$117M (5.2%)

Net Income
Before Taxes \$139M (6.1%)

Net Income
After Taxes \$85M (3.8%)

Retained
Earnings \$85M (3.8%)

Harsh Realities

- 23% of software projects fail to deliver any working software at all
- Of projects that do deliver, they average
 - 63% late
 - 45% over budget
 - 67% of features and functions
- 40% of commercial applications of computers have been uneconomical

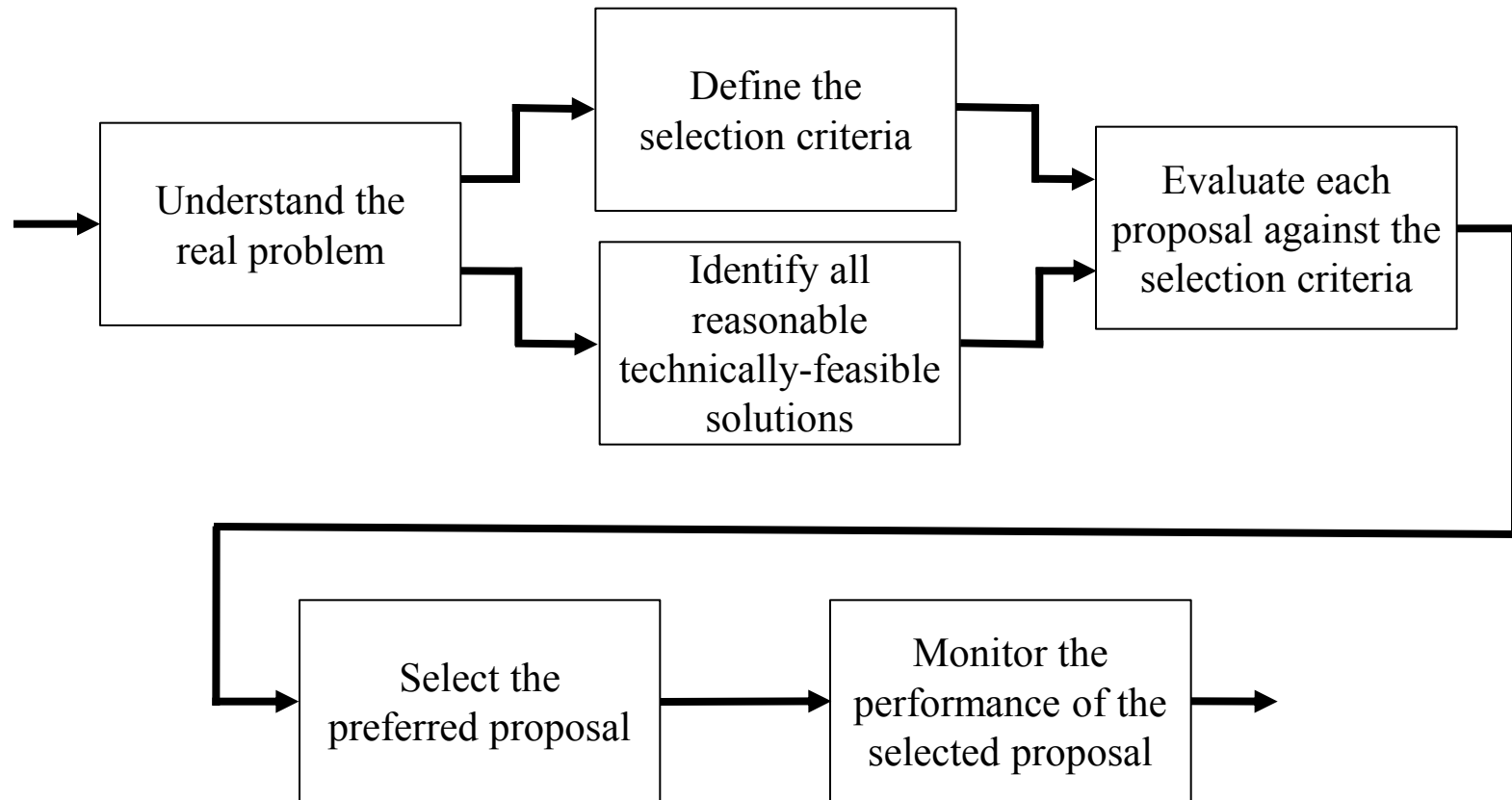
The Cost of Bad Decisions

- Poor software project performance can almost always be traced to bad decisions either by customer, project team, or both
 - Which projects to do
 - Not getting good requirements
 - Not giving good requirements
 - Using inappropriate technology
 - Choosing wrong design or architecture
 - Not giving project team adequate resources
 - Not planning and/or managing project
 - Not paying attention to quality
 - ...

Economics: the Science of Choice

*“... software economics has often been misconceived as the means of estimating the cost of programming projects. But economics is primarily a **science of choice**, and software economics should provide methods and models for analyzing the choices that software projects must make.”*

Business Decision-making Process



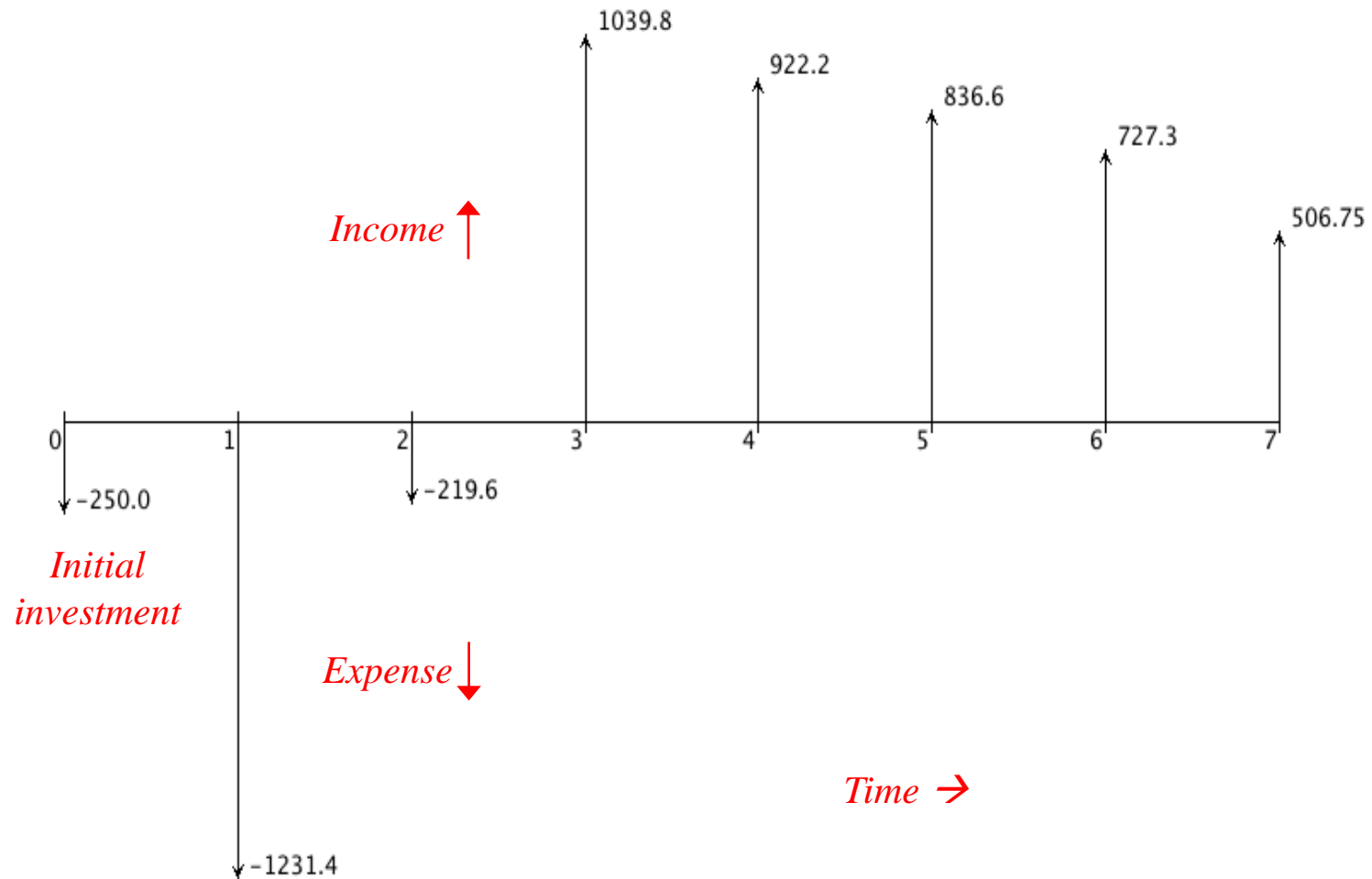
Categories of Cash-Flow

- Initial investment
 - One-time, non-recurring costs associated with starting up proposal
- Operating and maintenance costs
 - Recurring costs after proposal is started, that continue through retirement
- Sales income
 - Direct income generated by proposal
- Cost avoidance
 - Profit from reduced costs instead of more sales
- Salvage value
 - Recoverable value in assets at end

Alpine Payroll Processing Example

Year	Initial investment	Staffing cost	New sales income	Upgrade income	Maintenance income	Training income	Net cash flow
0	-\$250	\$0	\$0	\$0	\$0	\$0	-\$250
1		-2025	750	8	12	24	-1231
2		-2025	1650	45	46	64	-220
3		-675	1500	73	78	64	1040
4		-675	1350	88	96	64	922
5		-675	1200	120	128	64	837
6		-675	1050	138	151	64	727
7		-675	900	75	167	40	507

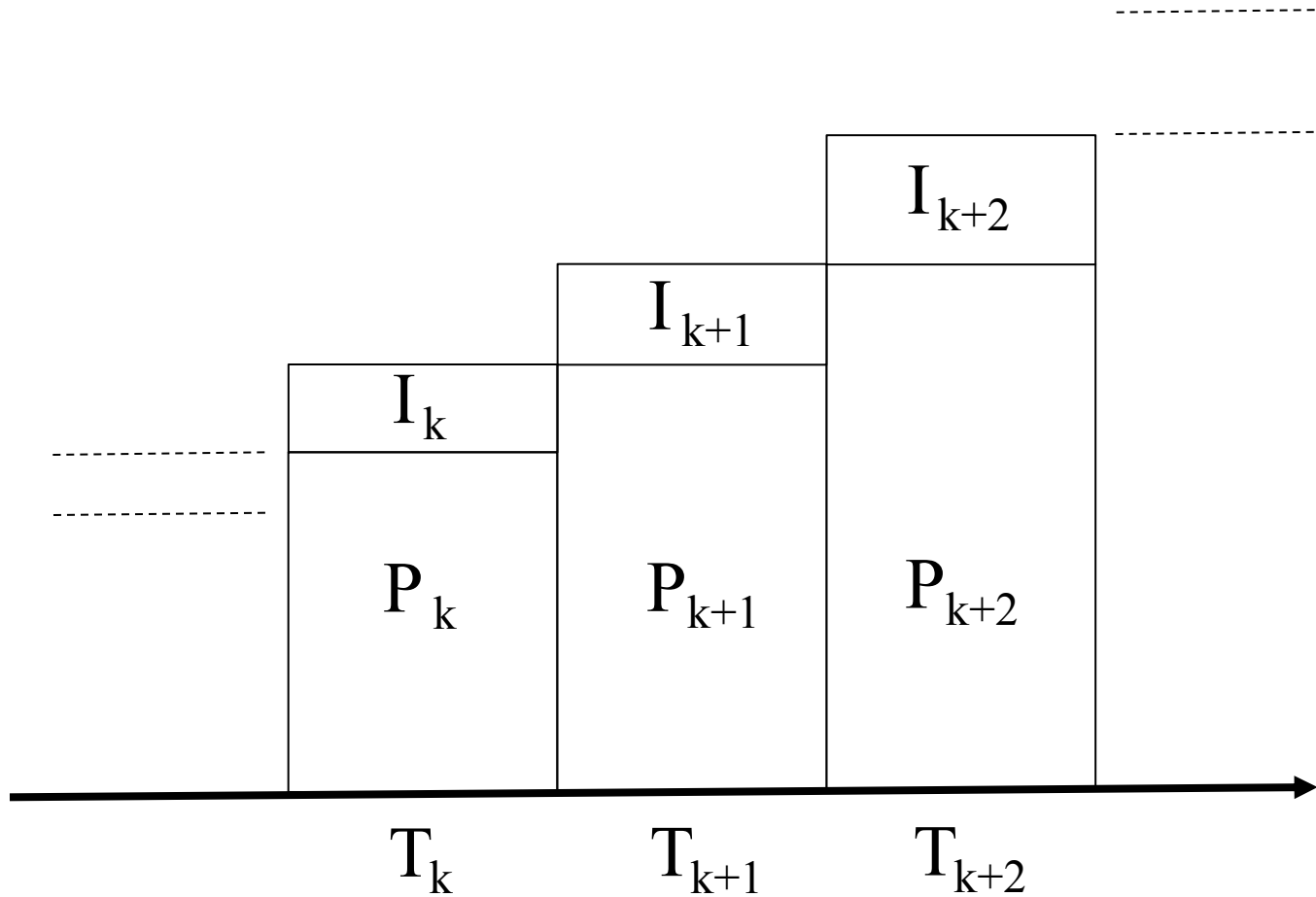
Cash-Flow Diagram



Time is Money

- Given amount of money at one time doesn't have same value as identical amount of money at a different time

Compound Interest



Present Worth, PW(i)

Year n	Net cash-flow at end of year	Present-worth factor (P/F,10%,n)	Equivalent value at end of year 0
0	-\$250K	1.0000	-\$250K
1	-1231K	0.9346	-1150K
2	-220K	0.8734	-192K
3	1040K	0.8163	849K
4	922K	0.7629	703K
5	836K	0.7130	596K
6	727K	0.6663	484K
7	507K	0.6228	316K
PW(7%)			\$1356K

Outline of Return on Software

- Part I: Introduction and Foundations
 - Proposals, cash flow, decision process, interest, equivalence, present worth, future worth, annual equivalent, internal rate of return, (discounted) payback period, ...
- Part II: Making For-profit Business Decisions
 - Alternatives, economic life, planning horizon, replacement, retirement
- Part III: Advanced For-profit Decision Techniques
 - Inflation, depreciation, accounting, income taxes
- Part IV: Making Decisions in Government and Non-profit Organizations
 - Benefit-cost analysis, cost-effectiveness analysis
- Part V: Present Economy
 - Break-even, optimization
- Part VI: Estimation, Risk, and Uncertainty
 - Estimation techniques, ranges of estimates, sensitivity analysis, expected value, Monte Carlo, decision trees, laplace, maximin, maximax, Hurwicz, minimax regret
- Part VII: Multiple Attribute Decisions
 - Value, measurement, dominance, satisficing, lexicography, non-dimensional scaling, additive weighting, analytic hierarchy process
- Part VIII: Summary



Key Points

- In the end it's all about making choices
 - Software professionals are faced with choices every day
 - Even apparently innocuous choices can have a noticeable effect on the organization's financial health
- As professionals, we need to make responsible choices
 - Choices that make sense both technically and to the organization

... software economics has often been misconceived as the means of estimating the cost of programming projects. But economics is primarily a science of choice, and software economics should provide methods and models for analyzing the choices that software projects must make. [Levy87]

- Using economic methods and models, software technical decisions can be aligned with goals of organization

References

- [Kidder81] Tracy Kidder, The Soul of a New Machine, Little, Brown & Co, 1981
- [Levy87] Leon Levy, Taming the Tiger – Software Engineering and Software Economics, Springer Verlag, 1987
- [Standish01] _____, CHAOS Chronicles, The Standish Group, West Yarmouth, MA, 2001
- [Tockey05] Steve Tockey, Return on Software, Addison Wesley, 2005

Contact Information

**Steve Tockey, MSE, CSDP
Chair, IEEE-CS PAB Certification Committee**

**Principal Consultant
Construx Software
10900 NE 8th St, #1350
Bellevue, WA 98004 USA**

SteveT@Construx.com