

The Incremental Funding Method

- A Data Driven Approach to Software Development

Mark Denne
Sun Microsystems Inc
mark.denne@sun.com

Jane Cleland-Huang
DePaul University
jhuang@cs.depaul.edu

Abstract

The last few years have seen intense scrutiny of the flawed business propositions underlying the dot.com bubble of the late 1990s. The prevailing attitude at that time was that investment in software could be repaid through the increased capital value of the company in expectation of future profits. The current IT environment is greatly changed. Not only are organizations no longer willing to invest in software development without clear expectations for returns, but they demand those returns in a much shorter time. This contrasts sharply with previous expectations, and provides the development community with a clear challenge to change the way we do business.

This paper introduces the Incremental Funding Methodology, a data-driven financially-informed approach to software development, characterized by its ability to analyze and sequence feature delivery, in order to maximize Net Present Value and to provide insight into the impact of development decisions on other financial metrics.

1. Software Development as a Value Creation Activity

In today's financially constrained IT industry, software development projects are unlikely to be funded unless they return clearly-defined, low-risk value to the business. Demands for shorter investment periods, faster time-to-market, and increased operational agility require new and radical approaches to software development that draw upon the expertise of both software and financial stakeholders [1]. Only by opening the traditional black box of software development to rigorous financial analysis, and by embracing software development as a value-creation activity, can organizations position themselves to maximize the returns on their software investments.

This paper describes the Incremental Funding Method [2]. IFM is a financially informed approach to software development, designed to maximize returns through delivering functionality in 'chunks' of customer valued features, carefully sequenced so as to optimize Net Present Value (NPV). IFM applies a financially rigorous analysis to the delivery sequence in order to compare alternate options and when necessary to change the dynamics of a project in order to secure executive buy-in and funding.

The initial IFM concepts were drawn from several years of experience in winning competitive contracts for systems integration and application development projects. To succeed in this highly competitive environment, the bid has to meet the budget, the development costs have to be low enough to ensure a reasonable margin for the bidder, and the margin must be justified against the risks. Clearly these ideas are not new and are true for any competitive procurement. However, as the IT industry continues to tighten its belt, and margins become progressively tighter, competitive differentiation cannot always be achieved through technical or price innovation. A different approach is necessary.

By optimizing the time at which value is returned to the customer, instead of concentrating only on controlling risk and cost, it is possible to present a uniquely differentiated value proposition even in circumstances that preclude traditional differentiation. By categorizing customer requirements in terms of units of value, it is often possible to sequence their development and delivery in such a way as to reduce initial investment costs, generate early revenue, and in the right circumstances to even transition a project to early self-funding status. Furthermore, the overall project cost is amortized into more manageable portions, each part of which has accountability for its returns. This is the essence of IFM.

IFM can be applied in conjunction with any iterative software development process such as the Rational Unified Process (RUP)[3] or eXtreme Programming (XP)[4]. The remainder of this paper discusses the principles and practices of IFM, illustrates its application through an example of an online Travel Agency application, and explains why IFM represents a financially responsible approach to software development.

2. The importance of cash flow

Let us first consider the hypothetical 5-year software development project depicted in Figure 1a. The project generates \$4M in revenue with costs of \$3.75M. A \$2.19M investment is required to bankroll the project, and the traditional ROI over 5 years is 11%. An early release of the software allows revenue to start flowing in year 4, although the real benefits are not experienced until year 5. On the cost side there are some early capital outlays associated with buying the hardware and software needed for development. A technology refresh cycle in year 4 updates the development hardware. There are also operational costs related to hardware and support, personnel, data center facilities fees, and marketing.

If however, it were possible to partition the application into distinct features, each having the ability to generate revenue, then the financial dynamics of the project could be significantly altered by early delivery of those features. If for example this project were deconstructed into four such features, all of equal value and brought to market in years 2, 3, 4 and 5 of the project, then the revenue would gradually build with year two experiencing revenue from feature one, year three from features one and two, and so on. Additional costs associated with packaging and separately releasing these features, including extra headcount costs to perform additional testing, are also accounted for in the financial model. The advantages of the incremental project over the traditional project can be summarized in net cash terms as follows:

- The project generates \$5M vs. \$4M over five years.
- The business invests \$1.11M vs. \$2.19M to fund the project.

Year	1	2	3	4	5	Total
Revenue				1,000	3,000	4,000
Hardware	-500	-100	-100	-200	-100	-1,000
Software	-300	-50	-50	-50	-50	-500
Headcount	-200	-300	-400	-300	-300	-1,500
Data Center	-30	-30	-30	-30	-30	-150
Sales/Mktg	0	0	-100	-200	-300	-600
Expense	-1,030	-480	-680	-780	-780	-3,750
Net Cash	-1,030	-480	-680	220	2,220	250
Investment	-1,030	-480	-680			-2,190
ROI						11%
DCF @ 10%	-936	-397	-511	150	1,378	-315

a. Traditional delivery

Year	1	2	3	4	5	Total
Revenue		500	1,000	1,500	2,000	5,000
Hardware	-500	-100	-100	-200	-100	-1,000
Software	-300	-50	-50	-50	-50	-500
Headcount	-200	-300	-400	-400	-400	-1,700
Data Center	-30	-30	-30	-30	-30	-150
Sales/Mktg	0	-100	-200	-300	-400	-1,000
Expense	-1,030	-580	-780	-980	-980	-4,350
Net Cash	-1,030	-80	220	520	1,020	650
Investment	-1,030	-80				-1,110
ROI						59%
DCF @ 10%	-936	-66	165	355	633	151

b. Incremental delivery

Figure 1 Impact of incremental feature delivery (\$US in thousands)

- The resulting ROI over five years is 59% vs. 11%.

Both approaches result in positive net cash positions and positive ROIs at the end of five years. However, it is important to take the time value of money into account when comparing cash flows. In line with traditional Discounted Cash Flow (DCF) analysis, the two different cash flows are discounted at a rate of 10%, to calculate their Net Present Value (NPV). In this example the first cash flow has a negative NPV while the second one has a positive NPV. This suggests that incremental feature delivery has the potential to transform an unprofitable project into a profitable one. By inference, it also has the potential to enhance the profitability of an already profitable project. The financial analysis of incremental feature delivery is the basis for the Incremental Funding Method (IFM) described in this paper.

3. The Incremental Funding Method

IFM decomposes the system into units of customer-valued functionality known as Minimum Marketable Features (MMF), and defined as small self-contained features that can be delivered quickly and that provide market value to the customer [2].

The practice of deconstructing a system in this way is not new. In his 1988 book on Software Engineering Management, Tom Gilb refers to an IBM Federal Systems Division experience of ‘LAMPS’, which was a 200 person-year project, delivered successfully over four years in 45 incremental deliveries [5]. More recently both the agile and more mainstream development communities have emphasized the importance of early delivery of executable functioning parts of the system [4][6]. In the same vein, the Standish Chaos report identified small project size as one of the most significant factors in project success [7]. By decomposing projects into MMFs, IFM therefore successfully reduces many of the risks inherent to large-scaled development.

An MMF’s value is typically measured in terms of both tangible and intangible factors such as revenue generation, cost savings, competitive differentiation, brand-name projection, and enhanced customer loyalty. MMFs are identified by customers, developers, and business stakeholders according to the adopted software development process. For example, in a RUP project, MMFs would be identified through top-down decomposition and then later further decomposed into use cases, whereas in an XP environment, MMFs are formed by clustering user stories into meaningful functional groups for delivery in single or sequential iterations. The MMFs from the Travel Agency example include:

- MMF A: Spot Booking – to reserve a single flight, car, or hotel.
- MMF B: Trip Booking – to reserve a complete trip including any combination of flights, cars, and hotels.
- MMF C: Itinerary Planning.
- MMF D: Frequent user discounts – to present special offers to frequent customers.
- MMF E: Proactive Notification – to send email notifications when discounts become available.

3.1 Incremental Architecture

Unfortunately, analyzing a system purely in terms of its value-generating features portrays only a partial picture. In many cases, MMFs can only be deployed once the underlying architectural infrastructure has been developed.

Although architecture has been defined in many different ways [8,9,10], these definitions all share the underlying assumption that an architecture defines the major ‘components’ or ‘elements’ of a system, and their interrelationships. This leads to the observation that the physical architecture of a system can be decomposed into deliverable elements in much the same way that its functionality can be deconstructed into MMFs [11,12]. We coin the term ‘Architectural Element’ (AE) to describe these elements. The online travel agency relies upon the following two AEs:

- AE 1: Messaging – to provide secure communications with system of record
- AE 2: Authentication System – to register and recognize frequent users.

3.2 Costs and Returns

Once the MMFs and AEs have been identified, their costs and projected revenues are analyzed over the number of periods established by the business. This task usually requires close collaboration between customers, marketing, and developers. Developers are responsible for estimating the cost and effort involved in developing each MMF and AE (collectively referred to as Elements from this point on). At the same time, the business stakeholders and marketers are responsible for estimating the revenue for each MMF. Figure 2, reports the outcome of this analysis, performed on a quarterly basis, for a project with a 3-year analysis period (12 quarters).

As an example, consider MMF D, which handles frequent user discounts. This feature is projected to cost \$500K to develop over two periods, and once delivered will start generating revenue through increased sales to prior customers. Initial revenue is projected at \$45K per quarter with gradual ramping up to reach \$180K per quarter by the third year. Risks are directly factored into this table, by reducing projected revenues according to the probability and impact of risk. For example, a base revenue projection of \$20K might be reduced

Element	Costs and Revenues per Period											
	1	2	3	4	5	6	7	8	9	10	11	12
AE 1	-200	0	0	0	0	0	0	0	0	0	0	0
MMF A	-200	-200	50	60	70	80	90	100	120	130	140	150
MMF B	-250	50	50	80	100	120	140	160	180	200	200	200
MMF C	-200	-200	80	112	144	176	208	240	272	304	320	320
AE 2	-200	0	0	0	0	0	0	0	0	0	0	0
MMF D	-250	-250	45	72	90	108	126	144	162	180	180	180
MMF E	-350	-350	35	70	105	140	175	210	245	245	245	245

Figure 2. Cost and Revenues for the Online Travel Agency (in \$US thousands)

by \$2.5K to \$17.5K, as a result of an identified 50% risk of losing one quarter of the revenue. ($\$20K \times 0.25 \times 0.5 = \$2.5K$).

Intangible values can also be factored into projected revenues using a number of techniques such as pair-wise comparisons to determine equivalency values [2], or through a more subjective process in which intangibles are directly quantified.

As purely revenue generating components, AE's are analyzed in terms of costs only. In some cases, it may not be necessary for the business to even incur capital costs for an AE, because a service may be included within the context of an ASP contract, perhaps as a result of outsourcing or hosting. In our example, both AEs will be developed in-house.

3.3 Precursors

In most applications, business and technology related dependencies exist between individual MMF and AE elements. In this simplified example of the travel agency, MMF's A, B, and C, are all dependent upon the transaction monitoring and messaging system, represented as AE 1, while MMFs D and E are dependent upon the authorization services of AE 2. Furthermore there is a linear dependency between MMFs A, B, and C, and another between MMFs D and E. These dependencies are summarized in the following precursor chains:

- AE 1 \leftarrow MMF A \leftarrow MMF B \leftarrow MMF C
- AE 2 \leftarrow MMF D \leftarrow MMF E

'Development dependencies' occur when a precursor must be developed prior to the development of its dependent element. "Delivery dependencies" represent a slightly looser form of dependency in which the elements may be

developed concurrently, but the dependent element cannot be deployed independently of its precursor.

3.4 Analysing Time Adjusted Delivery Options

Due to resource limitations and precursor relationships it is normally infeasible to develop all the elements during a single period, which introduces the need for sequencing. To examine the impact of sequencing decisions we need visibility into the way that MMF returns are affected by potential development schedules. This is accomplished through applying the discounted cash flow techniques outlined in Section 2, and calculating the NPV of each element for each starting period. The resulting Sequence-Adjusted NPVs (SANPVs) calculated at a discount rate of 1% per quarter, are shown in the non-shaded rows of Figure 3. For example the SANPV for MMF A if developed in period 1 is \$450K, whereas if it were developed in period 4 it would be reduced to \$106K.

The objective of IFM is to identify the delivery sequence that optimizes NPV within the constraints of both the precursor relationships and organizational factors such as the availability of upfront funding and other resources.

3.5 Assembling the data to drive the IFM Heuristic

In small projects with only a few elements, it is clearly possible to calculate the NPV of each feasible sequence in order to categorically determine the best one. However, because the number of feasible delivery sequences grows exponentially with the number of elements, this approach quickly becomes impractical. The point of impracticality is

Strand	1	2	3	4	5	6	7	8	9	10	11	12
1	-196	-192	-188	-185	-181	-178	-174	-171	-167	-164	-161	-158
	-196	-192	-188	-185	-181	-178	-174	-171	-167	-164	-161	-158
2	-196	-192	-188	-185	-181	-178	-174	-171	-167	-164	-161	-158
	-196	-192	-188	-185	-181	-178	-174	-171	-167	-164	-161	-158
B	1,013	838	667	500	351	220	108	13	-65	-125	-162	-197
	1,013	838	667	500	351	220	108	13	-65	-125	-162	-197
A.	450	326	211	106	11	-66	-134	-194	-244	-285	-319	-158
	383	277	179	90	10	-56	-114	-165	-207	-243	-271	-134
C.	1,450	1,174	904	651	428	234	69	-69	-179	-262	-319	-158
	1,232	998	768	553	364	199	58	-58	-152	-223	-271	-134
D.	603	452	304	159	31	-81	-177	-257	-321	-371	-398	-197
	513	385	259	135	26	-69	-150	-218	-273	-315	-338	-168
E.	765	561	361	164	-29	-190	-322	-424	-497	-541	-557	-276
	651	477	306	139	-24	-162	-274	-360	-422	-460	-474	-235
1A.	129	19	-82	-173	-247	-312	-368	-415	-453	-483	-319	-158
	91	13	-58	-121	-173	-218	-257	-290	-317	-338	-223	-110
2D.	256	112	-29	-154	-262	-354	-431	-492	-538	-562	-358	-158
	179	79	-20	-108	-183	-248	-302	-344	-377	-394	-251	-110
A.B	1,118	825	562	326	119	-53	-199	-319	-406	-483	-319	-158
	782	578	393	228	83	-37	-139	-223	-284	-338	-223	-110
BC.	2,187	1,742	1,318	927	585	289	39	-166	-327	-444	-319	-197
	1,531	1,219	923	649	409	202	27	-116	-229	-311	-224	-138
1A.B	630	369	138	-66	-235	-377	-493	-576	-650	-483	-319	-158
	346	203	76	-36	-129	-207	-271	-317	-357	-265	-175	-87
D.E.	964	617	276	-31	-291	-505	-673	-798	-879	-647	-398	-197
	530	339	152	-17	-160	-278	-370	-439	-483	-356	-219	-108
2D.E.	420	84	-219	-476	-686	-851	-972	-1049	-814	-562	-358	-158
	168	33	-88	-190	-274	-340	-389	-420	-326	-225	-143	-63
A.BC.	1,769	1,253	796	395	50	-232	-461	-638	-563	-483	-319	-158
	707	501	318	158	20	-93	-184	-255	-225	-193	-127	-63
1A.BC.	1,057	603	206	-135	-413	-639	-812	-734	-650	-483	-319	-158
	264	151	52	-34	-103	-160	-203	-183	-162	-121	-80	-39

**Figure 3. SANPV and Weighted SANPV for each Strand (In \$US thousands)
(Weighted SANPVs are shown in the shaded rows)**

reached surprisingly quickly. For example, even with average densities of precursor relationships, a typical project with 25-30 Elements has so many potential delivery sequences that it requires an unfeasibly long time to compute them all.

To address this problem we introduce the IFM Heuristic, which simply and quickly identifies a sequence that typically delivers within 92% of the optimal NPV of a project¹. Manual intervention is

required however when MMFs exhibit time sensitivity in their time-to-market constraints. In these cases additional comparisons must be made to select the optimal delivery sequence.

IFM starts by viewing sequencing options in terms of strands of Elements, where a strand is defined as a sequence of Elements linked by precursor relationships. Individual MMFs or AEs are special cases of strands. In this example, feasible strands would be, **1, 1A, 1AB, 1ABC, A, AB, ABC,**

¹ Summarized results from an empirical study of IFM are available online at <http://www.softwarebynumbers.org>

B, BC, C, as well as similar strands created from the path 2-D-E. Although not shown in this example, a single MMF may have multiple precursors and multiple dependencies. However, for the sake of simplifying this example, we illustrate this case study without multiple dependencies. To represent the fact that certain elements take more than one period to develop, we introduce the notation that a period (‘.’) represents an additional development period. MMF A, which takes two periods to develop, would therefore be represented as ‘**A.**’. As a further matter of convention, strand names are shown in bold text.

The next step in the process is to calculate SANPVs for each strand, representing the combined NPV for all elements in the strand if construction is started in a given period. For example:

$$\begin{aligned} \text{SANPV}(\mathbf{1A.B}, \text{Period 1}) &= \text{SANPV}(\mathbf{1}, \text{Period 1}) + \text{SANPV}(\mathbf{A.}, \text{Period 2}) \\ &\quad + \text{SANPV}(\mathbf{B}, \text{Period 4}) \\ &= \$-196\text{K} + \$326\text{K} + \$500\text{K} \\ &= \$630\text{K} \end{aligned}$$

As a strand that delivers \$100K in two periods of development is clearly more interesting than one that delivers \$100K in three periods, we also need to negatively weight the strands according to their lengths, and use the weighted SANPVs to drive the sequencing process. IFM applies the following weighting formula:

$$\text{Weighted SANPV} = \text{SANPV} \times (1 - (\text{Weighting Factor} \times (\text{Strand length in periods} - 1)))$$

This has the effect of promoting shorter more valuable strands and demoting longer ones during sequencing. The shaded rows in Figure 5, display the weighted SANPV figures (WSANPV). Empirical results suggest the use of a weighting factor between 12 and 20%¹. In this example, a weighting factor of 15% was applied to calculate the weighted SANPV of **1A.B** for period 1 as follows:

$$\begin{aligned} \text{WSANPV}(\mathbf{1A.B}, \text{Period 1}) &= \$630\text{K} \times (1 - (0.15 \times (4-1))) \\ &= \$346.50 \end{aligned}$$

3.6 The IFM Heuristic in Operation

The IFM Heuristic starts by examining the available strands for the first period and selecting the one with the highest WSANPV. A strand is

deemed to be “available” for selection if it has no unfulfilled precursors. In this case, therefore only strands starting with AEs 1 or 2 may be considered. The strand **1A.B** with a WSANPV of \$346,000 is therefore selected and AE 1 is developed in period 1. If concurrent development is supported, additional AEs or MMFs can be selected at this time.

The heuristic then proceeds to period 2 and performs the same comparison. Depending on the WSANPVs of the available strands in period 2, it may abandon the **1A.B** strand, or it may continue with it. In this case it continues because the strand **AB.** has the highest WSANPV of all available strands in period 2 (\$578K). This process is repeated until all the development periods are exhausted or no more available strands with positive WSANPVs remain.

In this way, IFM takes an iterative and risk-driven approach to NPV optimization. At the start of each iteration, stakeholders reexamine project risks, and cost and revenue estimates. If significant changes have occurred, the IFM sequencing algorithm is reexecuted, and a new sequencing plan developed.

In our example, assuming linear development of no more than one element per period and no significant changes in risks, costs, and revenue projections, then the selected sequence would be **1A.BC.** The IFM heuristic terminates the project after constructing MMF C because in period 7 there are no remaining available strands with positive WSANPVs. In other words, further development within the allocated development and analysis window would be counter-productive for the project in financial terms.

This illustrates an additional characteristic of IFM. In seeking to optimize project NPV it may discard some elements completely, simply because they not sufficiently profitable within the allocated analysis window. If the unscheduled features are deemed critical to the long-term success of the business then several options are available. First, the number of periods under analysis could be increased, second, an additional project cycle could be initiated with its own allocation of development and analysis periods, or third, concurrent development of multiple elements could be considered.

Figure 4 analyzes the financial metrics for the selected sequence and shows that the project

Sequence	Element	1	2	3	4	5	6	7	8	9	10	11	12	Net
1A.BC.	1	-200	0	0	0	0	0	0	0	0	0	0	0	-200
	A		-200	-200	50	60	70	80	90	100	120	130	140	440
	•													
	B				-250	50	50	80	100	120	140	160	180	630
	C					-200	-200	80	112	144	176	208	240	560
	•													
Cash		-200	-200	-200	-200	-90	-80	240	302	364	436	498	560	1,430
Net Cash		-200	-400	-600	-800	-890	-970	-730	-428	-64	372	870	1,430	
Cashpos status											X			10.15
Investment		-200	-200	-200	-200	-90	-80							-970
ROI														147%
Self fund status								X						
PV @	1.00%	-198	-196	-194	-192	-86	-75	224	279	333	395	446	497	1,232
Net PV		-198	-394	-588	-780	-866	-941	-718	-439	-106	289	735	1,232	
Breakeven status											X			10.27

Figure 4. Financial Analysis of the Travel Agency Project (in \$US thousands)

sequenced by the IFM Heuristic has the following characteristics:

- It generates just over \$1.4M in cash.
- It costs only \$970K to fund and becomes self-funding in Quarter 7.
- It delivers an ROI of 147% over three years.
- The NPV of the project at the discount rate defined earlier is just over \$1.2M
- In discounted cash flow (DCF) terms, it is profitable after quarter 10.

A brute force analysis of all feasible options indicates that the identified sequence is in fact the optimal one in terms of both ROI and NPV. This is not entirely surprising considering the small size and tight precursor constraints of this problem. However, empirical studies reported in <http://www.softwarebynumbers.org> show that for a set of typical MMFs (i.e. those not demonstrating significant time to market sensitivities) the heuristic normally finds a solution within 92% of the optimal.

3.7 Time-to-market Sensitive MMFs

Certain MMFs are particularly sensitive to time-to-market pressures and perform significantly better if delivered either prior to, or after an event has occurred. IFM handles these situations in two ways. First, alternate cost and revenue projections are

made according to an MMF's selected development period. Secondly, time-sensitive MMFs are 'flagged' for special attention. Once the IFM heuristic identifies a potential sequence, additional calculations are performed to determine if promoting or demoting the sensitive MMF within the sequence could improve the project's financial returns.

4. Analyzing the Financial Viability of a Project

From a financial perspective an incrementally funded software project passes through several phases depicted in Figure 5. Initially, the project receives funding to underwrite its development costs. After some time a successful project reaches a 'self-funding' point at which it starts to generate sufficient income to cover its ongoing development or maintenance costs. As the momentum builds and additional features are released, the project generates sufficient revenue also payback the initial investment. This milestone is referred to as breakeven status. From this time on, all additional revenue earns profit for the organization. Of course not every project behaves in such a textbook manner, but these milestones describe the type of financial behavior that we would expect to find in a successful project.

4.1 The Impact of Concurrent Development

At times an organization may need to develop MMFs concurrently in order to meet a completion deadline. In this section we consider the impact of such a decision by comparing a linear delivery sequence with a concurrent one. To accommodate this analysis for the Travel Agency example, the development and analysis periods are extended to 16 periods.

In this case, the optimal linear sequence is given as **1A.BC.2D.E.**, and the optimal concurrent sequences as **1A.BC.** and **2D.E.**, meaning that AEs 1 and 2 are developed in period 1, MMFs A and D in periods 2 and 3 (because they both require two periods), MMFs B and E in period 4, MMFs C and E in period 5, and finally MMF C is completed in period 6. Applying a financial analysis to these two delivery options results in the following observations:

- The concurrent sequence returns a higher NPV of \$4,716K vs. \$2,817K than the linear sequence.
- The concurrent sequence requires \$2,173K in upfront funding, vs. \$970K for the linear sequence.
- The concurrent project returns an ROI of 404% over four years vs. 296% for the linear project
- The concurrent project reaches self-funding (period 7) and break-even (period 12) status

later than the concurrent project (period 6 and period 11).

These results are interesting because they illustrate the tradeoffs that exist between concurrent and sequential development. Concurrent development may be attractive because it enables the development of more features and it increases NPV, however it also requires a higher initial investment and reduces the ROI of the project.

Clearly these types of tradeoffs must be carefully considered in light of the goals, objectives, and resources of an organization. Hybrid solutions can be used to reduce the initial funding costs to a supportable level, while attempting to maintain a satisfactory project-level NPV. For example, introducing the second sequence of **2D.E.** in period 5 instead of period 1, results in an NPV of \$3,408K, and an initial funding requirement of \$1,433K, performing midway between the concurrent and linear approaches. The trade-off is that it takes longer to achieve break-even and self-funding status than either of the other approaches. In this example self-funding status is reached in period 9, and breakeven status in period 12.

4.2 Getting a Project Funded

IFM creates a project environment in which all decisions, such as the concurrency one, can be

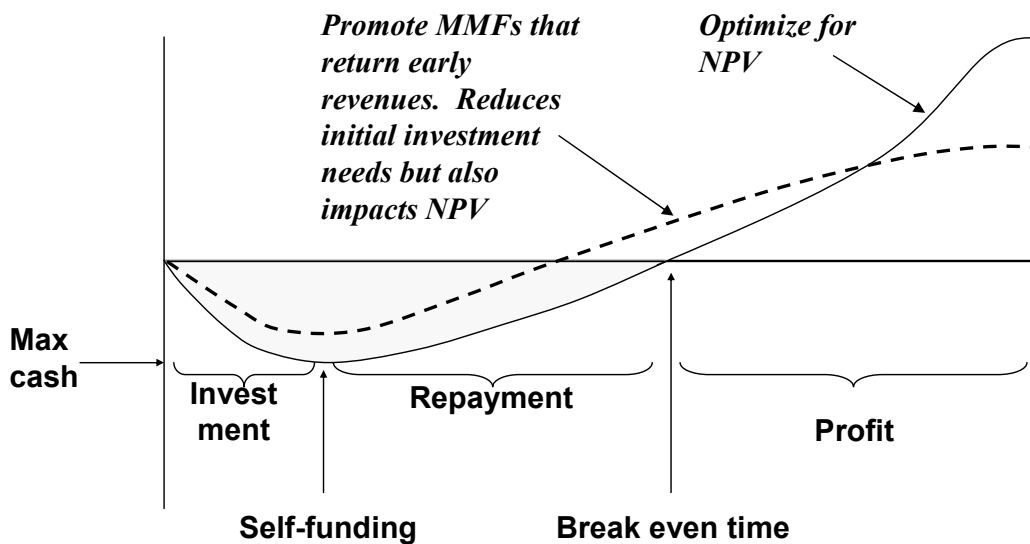


Figure 5. Manipulating Financial Metrics through Resequencing MMFs.

financially informed. There are times when a project is hard to fund simply because it requires a larger initial investment than an organization is willing or able to commit to. In these cases, MMFs and AEs can be re-sequenced in an attempt to change the shape of the cash flow diagram. For example, the initial funding requirements of a project may be reduced simply by identifying and bringing forward those MMFs that provide quick returns. The IFM heuristic normally favors MMFs that deliver high NPV, but the same heuristic applied against MMFs with a reduced view of the number of revenue periods, can identify those MMFs with earlier returns.

5. Conclusions

This paper has described an approach to software development in which financially informed decision making is an integral part of the requirements prioritization process. Only by recognizing the critical impact that development decisions have upon the financial well-being of a project can organizations hope to be successful in their software development endeavors. Furthermore, for software projects to contribute true value to an organization, developers can no longer afford to isolate themselves from the business side of an organization. Software development must truly be seen as a collaborative activity.

As businesses are increasingly forced to focus on their core competencies, they are increasingly looking to outsourcing as a viable option. This introduces the subsequent challenge of finding better pricing and payment options. IFM deconstructs an application into deliverable features that not only reduce the risks of large scaled development, but also provide a natural framework for establishing staged payments. This is beneficial to both the systems' integrator and to the customer.

The same model can be extended to Application Service Providers (ASPs). IFM equips the ASP with information needed to determine a cost structure that is beneficial to all parties involved. Both the ASP and the customer clearly benefit from the enhanced visibility into the financial and technical progress of the project. IFM's incremental nature provide early warning signs when projects start to go wrong, enabling midcourse corrections, and preventing the kinds of project death march

described by Yourdon [14] and documented within the Standish Chaos report [7].

Our current work in IFM involves the completion of a more extensive empirical study, and the documentation and assessment of industrial case studies in which IFM has been applied. Further information about IFM and an ongoing discussion related to these concepts, can be found at <http://www.softwarebynumbers.org>.

References

- [1] B. Boehm and K.J. Sullivan, "Software Economics: A Roadmap, 22nd International Conference on Software Engineering, June, 2000, pp. 937-946.
- [2] Mark Denne and Jane Cleland-Huang, *Software by Numbers, Low-Risk, High-Return Development*, Prentice-Hall, 2003.
- [3] P. Kruchten. *The Rational Unified Process, An Introduction*. 2nd ed. Addison Wesley, 2000.
- [4] Kent Beck, *Extreme Programming Explained, Embrace Change*, Addison Wesley, 1999.
- [5] Tom Gilb, *Principles of Software Engineering Management*, Addison-Wesley, 1988.
- [6] Per Kroll and Phillippe Kruchten, *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*, Addison-Wesley, 2003.
- [7] The Standish Group, *Chaos Report*, 1995, <http://www.standishgroup.com>
- [8] L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice*, Addison-Wesley 1997
- [9] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*, Addison-Wesley, 1998.
- [10] M. Shaw and D. Garlan, *SoftwareArchitecture: Perspectives on an Emerging Discipline*, Prentice Hall, 1996.
- [11] Jane Cleland-Huang, Mark Denne, and David Schmelzer, "The Incremental Funding Methodology, An Agile Approach to Software Architecture", Workshop on Requirements Engineering for Adaptive Architectures, Monterey Bay, CA, Sept. 2003.
- [12] Kenn Busse and David Hecksel, "Applying an Architecture Centric Iterative Development

Process to the Java Platform”, Java One Proceedings, 2001.

- [13] J. Karlson and K. Ryan, “A Cost-Value Approach for Prioritizing Requirements”, *IEEE Software*, 14 (5):67-74, 1997.
- [14] E. Yourdon, *Death March*, Prentice-Hall, 1999.