

A GESTÃO DE RECURSOS NO DESENVOLVIMENTO DE SISTEMAS COMPUTACIONAIS

Antonio Carlos Tonini
Carlos Hideo Arima

Resumo:

O dia-a-dia da gerência de desenvolvimento de software ocorre em ambiente bastante instável, onde a demanda dos recursos humanos e computacionais sofre uma concorrência acirrada entre os vários projetos que estão sendo desenvolvidos e mantidos. A literatura especializada, em geral, interpreta essa realidade de forma estática, fazendo recomendações para a elaboração de estimativas e controle, não aprofundando os estudos na questão da complexidade do gerenciamento. Este estudo pretende contribuir com a questão, analisando um dos aspectos da gerência do desenvolvimento de sistemas computacionais que é o da alocação de recursos, a partir de uma visão dos custos envolvidos nessa atividade.

Área temática: *A Gestão de Custos e os Sistemas de Informação*

VIII Congresso Brasileiro de Custos

**A GESTÃO DE RECURSOS NO DESENVOLVIMENTO DE SISTEMAS
COMPUTACIONAIS**

AUTORES:

ANTONIO CARLOS TONINI
Universidade Ibirapuera
Rua Major Freire, 594 ap.32 – Vila Monte Alegre
Cpf: 049.597.368-87
CEP 04304-110
São Paulo - SP
actonini@uol.com.br

CARLOS HIDEO ARIMA
Universidade Ibirapuera (São Paulo – SP)
Unisinos (São Leopoldo – RS)
Rua Machado Bittencourt, 205 conj 35 – Vila Clementino
São Paulo – SP – cep: 04044-010
info@arimaconsulting.com.br

Telefones: (11) 5589-9260 res
(11) 9292-3148 cel
(11) 5080-3575 recados
(11) 5574-5524 fax

Resumo

O dia-a-dia da gerência de desenvolvimento de software ocorre em ambiente bastante instável, onde a demanda dos recursos humanos e computacionais sofre uma concorrência acirrada entre os vários projetos que estão sendo desenvolvidos e mantidos. A literatura especializada, em geral, interpreta essa realidade de forma estática, fazendo recomendações para a elaboração de estimativas e controle, não aprofundando os estudos na questão da complexidade do gerenciamento.

Este estudo pretende contribuir com a questão, analisando um dos aspectos da gerência do desenvolvimento de sistemas computacionais que é o da alocação de recursos, a partir de uma visão dos custos envolvidos nessa atividade.

PALAVRAS-CHAVE: Gerenciamento, Prazos, Custos, Desenvolvimento de Sistemas, Alocação de Recursos, Otimização, Aceleração

A GESTÃO DE RECURSOS NO DESENVOLVIMENTO DE SISTEMAS COMPUTACIONAIS

INDICE

1. Introdução	2
2. Características do desenvolvimento de sistemas computacionais	3
3. A distribuição do tempo no desenvolvimento de sistemas computacionais ..	4
4. O processo de gerenciamento	5
5. Os custos do desenvolvimento de sistemas computacionais	6
5.1. Custos diretos ou internos	7
5.2. Custos indiretos ou externos	10
5.3. Custo total	10
6. A alocação de recursos	11
6.1. Nivelamento dos recursos	12
7. O problema da aceleração	13
8. Conclusões	15

1. Introdução

Em boa parte das empresas, a área de desenvolvimento de sistemas computacionais não consegue atender plenamente as expectativas dos usuários. Uma das maiores queixas dos usuários é quanto aos prazos de desenvolvimento. No seu entender os sistemas nem sempre estão disponíveis quando são necessários, são entregues fora do prazo combinado – antes ou depois.

Depois de quase meio século de vida desta atividade, os maiores responsáveis por esse problema continuam os mesmos: os usuários, os desenvolvedores e os gerentes. Os esforços no desenvolvimento de metodologias de desenvolvimento, ferramentas de produtividade e sistemas corporativos têm contribuído decisivamente para o bom desempenho dos desenvolvedores e o atendimento dos requisitos dos usuários.

A atividade gerencial, no entanto, continua sendo, segundo os especialistas, a causa principal e crônica dos problemas de efetividade desta atividade. Por esta razão, os sistemas de gerenciamento da qualidade voltados às atividades da tecnologia da Informação (TI), mais respeitados no mundo, tais como as normas *ISO 15504 - SPICE* e os modelos *CMM, CMMI* e *PMI* são unânimes em eleger o gerenciamento como o elemento chave mais importante. Essa ênfase acontece porque o gerenciamento envolve a análise de alternativas e a tomada de decisão, que muitas vezes ocorrem em cenários não tão estáveis.

As metodologias de gerenciamento são apresentadas e discutidas pela literatura em geral, considerando um ambiente estável, formado por um único projeto, havendo disponibilidade infinita dos recursos.

Entretanto, o dia-a-dia das áreas de TI ocorre num cenário bastante agitado, onde os projetos concorrem pelo uso dos recursos e estes são, via de regra, bastante escassos. A postura gerencial assume, então, uma dimensão fundamental para garantir o sucesso de cada desenvolvimento, como se ele fosse o único e o mais importante da organização, sem a concorrência ou escassez dos recursos.

Este estudo analisa um dos aspectos da gerência do desenvolvimento de sistemas computacionais que é o da gestão de recursos, a partir de uma visão dos custos envolvidos nessa atividade.

2. Características do desenvolvimento de sistemas computacionais

O desenvolvimento de sistemas computacionais é uma atividade que depende da iteração de idéias e interpretações pessoais. Uma das suas características mais marcantes, é que existe uma incerteza quanto ao resultado, tanto em relação ao produto final quanto sobre o prazo de entrega.

Segundo Page-Jones [09], em muitas organizações, a cultura de *software* baseia-se, ainda, em crenças absurdas, que se espalham por todos os níveis da organização. De uma forma geral, acredita-se que os investimentos típicos em ferramentas de produtividade e em metodologias de gerenciamento são mera burocracia ou, então, caros demais. Acreditam ainda que, se utilizados, contribuem para o alongamento dos prazos de desenvolvimento.

Esta crença errônea é agravada pela limitada cultura de gerenciamento de usuários e clientes, que tendem a pressionar por prazos completamente fora da realidade. DeMarco [03] afirma que, *“regularmente pondo o processo de desenvolvimento sob extrema pressão de tempo e depois aceitando produtos de baixa qualidade, a comunidade de usuários mostra seu verdadeiro padrão de qualidade”*.

De uma forma geral, segundo DeMarco [03], os gerentes de projetos não estão acostumados a fazer estimativas de tempo. Quando o fazem, costumam basear-se nas estimativas passadas, mesmo sabendo que elas podem estar incorretas; e, ainda mais grave, não sabem precisar o quanto estão incorretas. Ainda há aqueles que se recusam a fazer novas estimativas por julgarem perda de tempo, uma vez que correm o risco de obterem resultados também incorretos e, portanto, estarem desperdiçando tempo. Os gerentes inexperientes tentam cumprir o prazo, a qualquer custo, dando a máxima velocidade na fase inicial. Não se preparam para os momentos de impasse, quando os ajustes são inevitáveis.

Ainda mais, a falta de liderança do gerente gera um descontrole emocional nas pessoas, redundando um excessivo otimismo da equipe, que acaba sendo irreal ou até um completo abandono, não-dedicação e não-comprometimento. É comum, acrescenta Hehn [06] que os projetos apresentem uma carga

adicional de estresse nas pessoas. Quando elas não são corretamente administradas, tendem a causar mais danos que todos os erros técnicos juntos.

Ainda segundo DeMarco [03], como raramente são registrados os prazos reais, ao final dos projetos, o que fica no inconsciente do gerente como regra de referência são os prazos que ele costuma prometer e não os que ele não cumpre. Nesta linha de raciocínio, parece que o cerne da questão é a ausência de dados históricos sobre o comportamento da atividade. Com esses dados em mãos, o gerente tem muito mais condições de intervir e gerar estimativas mais confiáveis.

3. A distribuição do tempo no desenvolvimento de sistemas computacionais

A forma como as organizações desenvolvem seus sistemas computacionais é bastante diferente para cada uma delas. Esta constatação decorre da influência das estruturas organizacionais, tamanho da empresa, importância da atividade no contexto do negócio, entre outras.

Portanto, o esperado é que o uso do tempo possa ser entendido de diferentes maneiras por diferentes organizações. Cada atividade consome uma determinada quantidade de tempo para ser realizada. Desta forma, deve-se saber como cada atividade “gasta” o tempo.

Nakajima [08], ao estudar os processos fabris, afirma que o tempo despendido nas atividades pode ser dividido da forma indicada no quadro abaixo:

<u>Divisão do uso do tempo</u>				
Tempo Total DISPONIBILIZADO				
Tempo APLICADO				Tempo de atividade PARADA
Tempo PRODUTIVO			Tempo IMPRODUTIVO	
Tempo de PROGRESSO	Tempo de MODIFICAÇÕES	Tempo de RETRABALHO		

Onde:

- **tempo total DISPONIBILIZADO** – corresponde ao tempo delimitado pelas datas de início e fim do projeto, durante o qual algum tempo foi APLICADO ao projeto e outro em que NADA se fez;

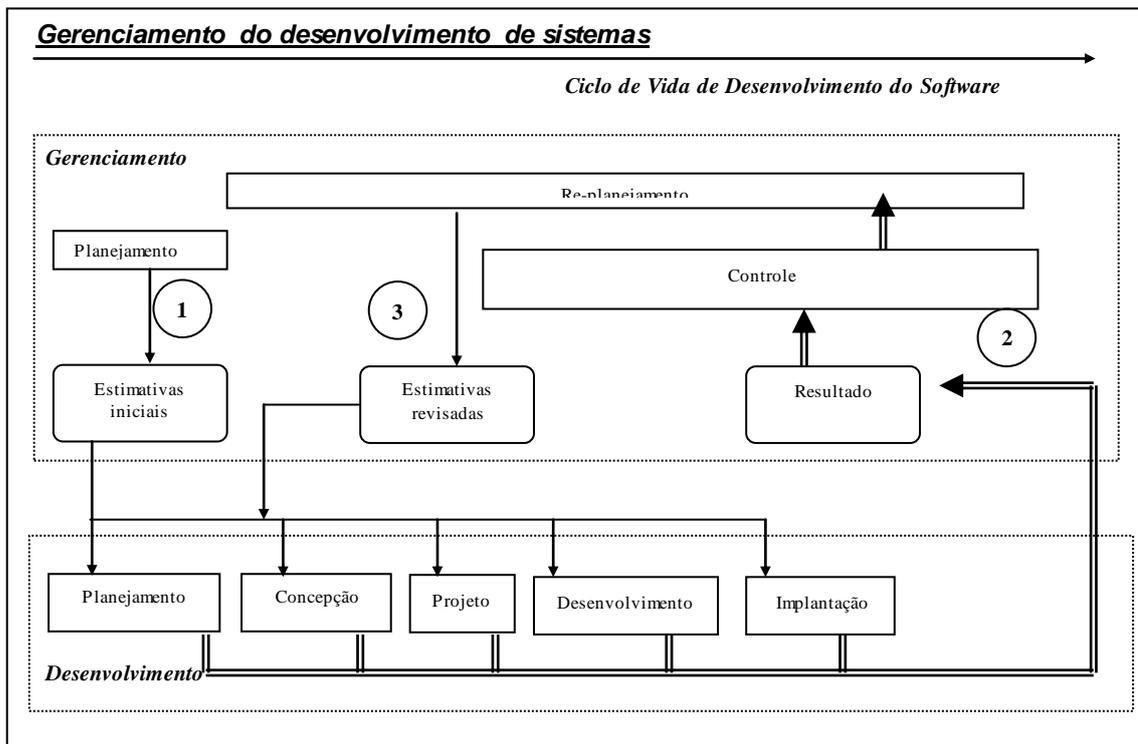
- **tempo de PARADA** – refere-se ao tempo durante o qual não houve trabalho relativo ao projeto. Equivale aos tempos de espera por uma tomada de decisão ou utilização do tempo para outras atividades;
- **tempo APLICADO** – constitui-se de todo tempo em que houve trabalho real de desenvolvimento de sistema (trabalhos técnicos, reuniões, etc.) e tempo despendido em atividades paralelas ao projeto que contribuem, até certo ponto, com o trabalho;
- **tempo IMPRODUTIVO** – é o tempo empregado em atividades paralelas ou complementares ao projeto, como estudo de material de apoio, treinamento em novas metodologias e tecnologias, consultas, preparação de materiais acessórios (também chamado de tempo de *setup*). Uma parte desse tempo, mesmo diminuindo a velocidade de desenvolvimento do projeto, pode significar ganho em outras atividades do mesmo projeto ou de outros;
- **tempo de RETRABALHO** – é a perda de tempo devido a trabalhos feitos de forma incorreta ou cujos resultados não se adaptam ao previsto;
- **tempo de MODIFICAÇÕES** – refere-se ao tempo em que não houve avanço, pois alguns dos resultados devem ser alterados; contudo, é um tempo que contribui para a consecução dos resultados;
- **tempo de PROGRESSO** – corresponde ao tempo empregado e que gera resultados de acordo com o esperado na atividade, podendo estar subdividido em:
 - **tempo de DESENVOLVIMENTO** – é aquele empregado na construção de NOVAS funcionalidades para o sistema aplicativo;
 - **tempo de MELHORIAS** – equivale ao empregado na reformulação ou otimização de funções já existentes, que trazem melhorias de *performance* do aplicativo. Normalmente as melhorias, assim como as modificações, consomem uma parcela de tempo para desfazer o que já existe.

Por outro lado, conforme DeMarco [03], “só se pode controlar o que se pode medir” e, portanto, é fundamental que haja um processo de apontamento do tempo utilizado e do trabalho que foi executado. Enfatiza ainda o autor que, deve haver alguma forma de se garantir a qualidade e a isenção de erros e manipulação do apontamento. Se possível, esta atividade deve ser mecanizada e automatizada.

4. O processo de gerenciamento

O gerenciamento do uso do tempo no desenvolvimento de sistemas computacionais é uma atividade que deve acompanhar o processo durante todo o seu ciclo de vida.

De uma maneira geral, a participação gerencial ocorre em três “momentos”, conforme procura mostrar o quadro a seguir:



O primeiro momento, ocorre nos instantes iniciais da vida de um software, quando é estabelecida uma estimativa de prazo de entrega. Segundo DeMarco [03], para aqueles que não fazem apontamento das entregas reais, esta estimativa está muito mais para uma intenção ou meta, do que um comprometimento sério fundamentado na disponibilidade de recursos e um consenso com os usuários.

A segunda participação gerencial ocorre no desenrolar dos trabalhos e significa colher dados do que realmente está ocorrendo. Segundo Page-Jones [09], o que deveria ser uma rotina gerencial, acaba sendo, quando feito, um atividade formal apenas na conclusão dos trabalhos. Pressman [10] afirma ainda que, quando ocorre, o *feed-back* muitas vezes é bastante impreciso, do tipo: o trabalho está em andamento... falta pouco... está quase pronto... pretende-se encerrar no prazo.

O terceiro instante é aquele que é resultado de uma análise dos fatos ocorridos. Como estes raramente ocorrem, ou porque o *feed-back* não permite grandes conclusões, a decisão gerencial acaba não acontecendo.

5. Os custos do desenvolvimento de sistemas computacionais

Conforme define Leone [07], os custos não são a causa do processo e, sim, a consequência do uso dos recursos. Cada tipo de custo deve ser traduzido numa unidade de medida monetária mais característica com o indicador

quantitativo que responde pela sua variação. Assim, como nos projetos de construção civil a unidade monetária é o valor do metro quadrado, no desenvolvimento de software é o valor hora do desenvolvedor, que deve levar em conta o quanto a organização remunera o recurso humano nos seus projetos computacionais.

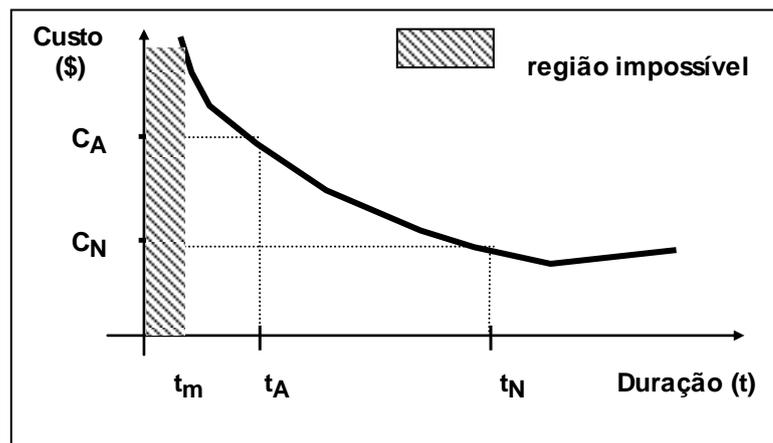
Os custos registrados pelo desenvolvimento de sistemas computacionais podem ser classificados em dois grupos, custos diretos ou internos e custos indiretos ou externos.

5.1. Custos diretos ou internos

São aqueles que dependem diretamente da execução das atividades de desenvolvimento. Eles se caracterizam pelo fato de deixar de existir sempre que a tarefa de desenvolvimento deixa de ser realizada. Dessa forma, cada unidade, cada programa, cada módulo ou etapa concluída de um sistema corresponde a um gasto relativo ao tempo que consumiram para a sua confecção. Assim, se não houver desenvolvimento, não há a ocorrência do custo direto ou interno.

Em geral, esse tipo de custo diminui com o aumento do tempo, dentro de certos limites, ou melhor, ele aumenta quando o tempo diminui. Isso se deve, em geral, ao fato de, para aumentar a velocidade da produção de software, necessita-se de mais homens/horas e de mais equipamentos trabalhando num dia. Ultrapassando certo limite, um aumento de recursos produz um aumento de custo, principalmente pela falta de possibilidade de usá-los convenientemente.

A curva custo direto x tempo tem a seguinte conformação:



Onde:

t_m = determina o tempo mínimo exigido pelas atividades de desenvolvimento de software, isto é, abaixo deste tempo não é possível a sua produção, determinando a região impossível.

t_A = tempo de desenvolvimento de forma acelerada. Esse é o ponto mínimo que se pode obter, empregando todos os recursos possíveis para acelerar a atividade. A partir desse ponto, um excesso de recursos aumenta o tempo, pela impossibilidade de usá-los convenientemente.

t_N = tempo normal (ótimo) de desenvolvimento do sistema. Acima deste tempo, a curva de custo direto x tempo tende a subir, pois trabalhar muito lentamente encarece qualquer atividade. Este ponto corresponde ao custo direto mínimo que se pode obter aumentando o prazo de execução dentro dos limites indicados pelo estágio atual de TI. Além desse prazo o custo subirá em resposta ao aumento do tempo por falta da eficiência do processo. Esse ponto pode ser chamado de duração normal da atividade.

C_A = custo correspondente ao tempo de desenvolvimento de forma acelerada

C_N = custo correspondente ao tempo normal (ótimo) de desenvolvimento do sistema.

R = É a razão entre a variação de custo ($C_A - C_N$) e a variação do tempo ($t_A - t_N$), representando o custo marginal de aceleração:

$$R = \frac{(C_A - C_N)}{(t_A - t_N)}$$

A divisão do uso do tempo nas atividades de desenvolvimento de sistemas redundam em custos diretos. Para alguns autores, entretanto, o tempo de parada ou o tempo improdutivo pode não ser caracterizado como um custo direto, uma vez que o fato de haver uma parada nos trabalhos, depende muito mais de condições organizacionais do que, propriamente dito, de um determinado sistema. Entretanto, no contexto desse trabalho, como o foco é o uso do tempo, considera-se esses usos de tempo como custos diretos. Assim os custos dependentes do tempo são os seguintes:

- **Custo de progresso e melhorias**

Conforme Garrison [05], esse custo representa o sacrifício que a organização paga para ter um sistema computacional implantado e ajustado à realidade dos negócios para o qual foi escrito. É composto exclusivamente do tempo despendido em atividades de desenvolvimento que resultam em produto de *software* que está em conformidade com as especificações.

- **Custo das modificações**

Conforme Fernandes [04], a instabilidade dos requisitos dos usuários provocada pela insegurança e pela dificuldade de visualizar as necessidades de forma sistêmica, agravada pela dificuldade de os desenvolvedores “falarem” a mesma linguagem dos usuários, é a grande causa dessas modificações. Consequentemente, acaba sendo uma das grandes responsáveis pelo não-cumprimento do prazo de entrega.

Para Arthur [01], as modificações sempre implicam a destruição e reconstrução parcial do *software*. Quanto mais modificações forem requisitadas, maior o tempo para a adaptação do *software*. Para o autor, nem sempre é possível medir com precisão esse tempo, visto que ele fica “disfarçado” como melhorias.

- **Custo do retrabalho**

Esse custo representa o tempo gasto para a correção dos erros de desenvolvimento. Quanto mais tarde for descoberto o erro, mais demorado e, portanto, mais cara fica a correção. Na opinião de Hehn [06], as pessoas normalmente preferem não falar de seus fracassos e erros, ficando esse tempo “mascarado” como desenvolvimento de progresso. Dependerá muito da capacidade gerencial a identificação e a minimização dessa situação junto a sua equipe.

- **Custo do tempo improdutivo**

O custo improdutivo representa o tempo gasto em atividades de treinamento, leitura de material de apoio, reuniões, consultoria. De acordo com Valeriano [11], esse tempo pode se traduzir em melhorias na capacitação da equipe de desenvolvimento. Page-Jones [09] reafirma essa posição, dizendo que esse tempo “perdido” no projeto pode significar ganhos proporcionalmente maiores em projetos futuros. Este custo pode absorver também o rateio da mudança tecnológica sobre o projeto.

O comportamento que melhor representa a evolução desse tipo de custo é um misto entre variável e fixo, afirma DeMarco [03]. Quanto maior e complexo o projeto, o comportamento será variável, pois explica um atraso na entrega. Quanto mais homogênea for a equipe e estável for o ambiente computacional, o comportamento tende a se fixar em uma média, influenciando pouco no cumprimento do prazo.

- **Custo do tempo parado**

Arthur [01] afirma que o tempo gasto na tomada de decisão é o que mais contribui para o não cumprimento do prazo. Quanto menos dependente dos níveis hierárquicos superiores for o desenvolvedor e o usuário, mais rapidamente o desenvolvimento do *software* acontece. Esse custo traduz, portanto, a perda ou ganho de tempo no projeto devido à estrutura

organizacional. O comportamento desse tipo de custo é aleatório, pois é bastante difícil controlar a variável *tomada de decisão*.

Valeriano [11], ao comentar esse caso, afirma que, geralmente, só se aponta o tempo parado dos desenvolvedores. Quase ninguém se preocupa em onerar esse custo com o valor do tempo do gerente de quem depende a decisão.

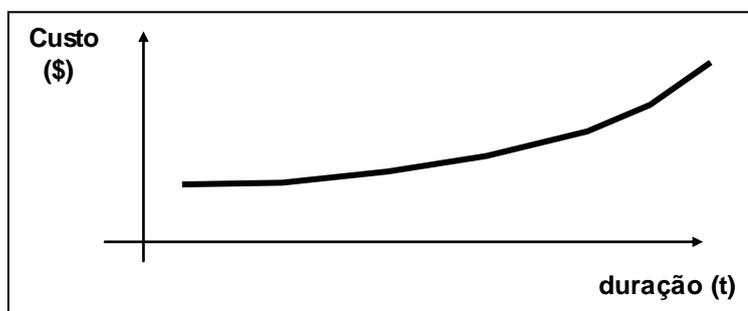
- ***Custo pela entrega antecipada***

Esse custo traduz o sacrifício que o usuário faz quando um *software* lhe é entregue antes do prazo esperado. A literatura sobre desenvolvimento de sistemas computacionais é farta na abordagem do atraso da entrega dos sistemas computacionais, sendo raras as citações sobre a entrega antecipada.

5.2. Custos indiretos ou externos

São todos aqueles que ocorrem inevitavelmente, mas que não dizem respeito diretamente à execução do desenvolvimento dos projetos sistêmicos. O comportamento esperado para este tipo de custo é que eles sempre tende a crescer com o passar do tempo. Em outras palavras, quanto mais o tempo passa a incidência dos custos indiretos em um sistema é cada vez maior.

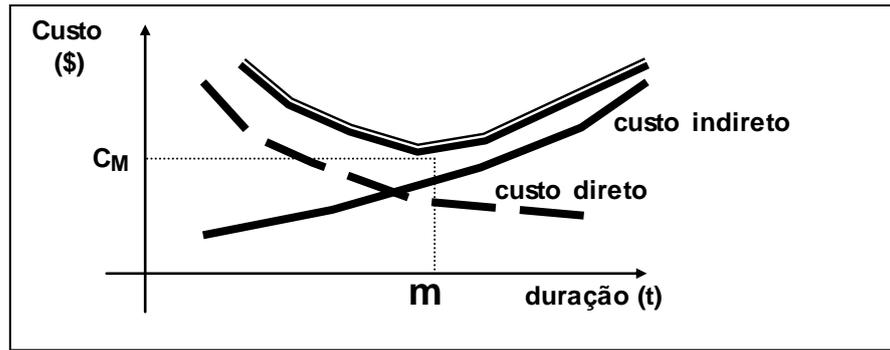
A representação gráfica para o custo indireto x tempo é a seguinte:



5.3. Custo total

O Custo Total de um projeto representa a soma dos valores de custo diretos e indiretos, podendo ser representado pela curva indicada no próximo quadro. Essa curva passa, em geral por um ponto **m** mínimo, que corresponde à duração do projeto (tempo ótimo), em que o custo é o menor possível, pois além desse ponto a influência do custo indireto se faz sentir mais fortemente e o custo total começa a crescer com o tempo.

A representação gráfica para o custo total x tempo é a seguinte:



6. A alocação de recursos

A alocação e o remanejamento dos recursos é a atividade gerencial mais crucial do processo de gerenciamento da atividade de desenvolvimento de sistemas computacionais, pois sendo os recursos escassos, o favorecimento de alguma atividade sempre ocasiona o prejuízo para alguma outra.

O custo de oportunidade do processo, deve contemplar esse aspecto.

A questão da realocação de recursos sempre ocorre uma vez que a demanda dos recursos normalmente não é uniforme e a ocorrência de fatos não planejados tende sempre a ser maior que os fatos e atividades que possam ser planejados.

Para visualizar melhor as ações gerenciais, é utilizado o seguinte exemplo:

Um único projeto, contendo 5 atividades:

BD1 – preparação do banco de dados 1;

BD2 – preparação do banco de dados 2;

PGM1 – elaboração do programa 1, que depende do BD1;

PGM2 – elaboração do programa 2, que depende do BD2;

PGM3 – elaboração do programa 3, que depende do BD1 e BD2.

Os tempos da rede PERT, bem como o cronograma, com as folgas é o representado pelo seguinte quadro:

Exemplo de Projeto																				
Atividade	Depende de:	Duração (semanas)	+ cedo		+ tarde		Folga	Cronograma												
			Ini	Fim	Ini	Fim		1	2	3	4	5	6	7	8	9	10			
BD1	-	3	1	3	2	4	1	█	█	█	█									
BD2	-	4	1	4	1	4	-	█	█	█	█									
PGM 1	BD1	2	4	5	9	10	5				█	█	█	█	█	█	█	█	█	█
PGM 2	BD 1 e 2	6	5	10	5	10	-													
PGM 3	BD 2	4	5	8	7	10	2													
Recursos	Administrador de dados							2	2	2	1									
	Programador										1	3	2	2	2	1	1			
	Computador							2	2	2	2	3	2	2	2	1	1			

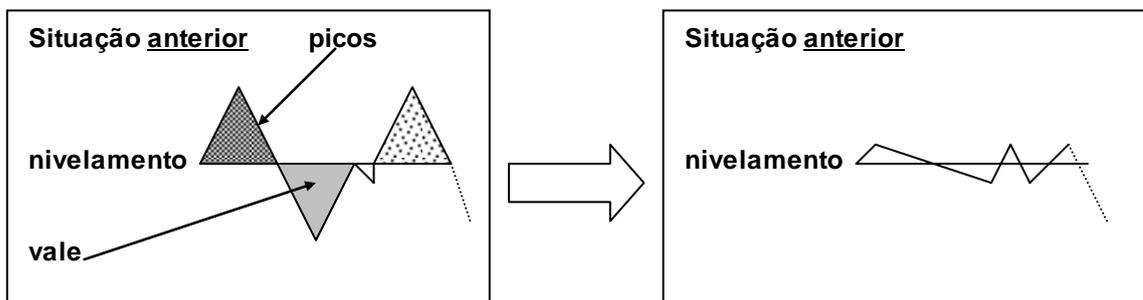


Neste exemplo:

- No 3ª semana, começa a ociosidade de um recurso “Administrador de dados”;
- Na 3ª semana, inicia a necessidade dos trabalhos de um “Programador”;
- Só na 5ª semana, há a necessidade de 3 programadores e 3 computadores. Nessa semana é a que apresenta a maior demanda de recursos.

6.1. Nivelamento dos recursos

O nivelamento de recursos deve ser feito toda vez que se detecte a ocorrência de picos ou vales no uso dos recursos e tem por objetivo normalizar a demanda desses ao longo do projeto. Resumidamente:

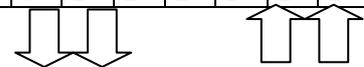


O deslocamento da execução das tarefas deve ser feito seguindo-se os seguintes critérios:

1. identificar as atividades que estão nos caminhos críticos dos projetos;
2. fixar para elas os seus recursos necessários;
3. identificar as demais atividades por ordem crescente de prioridade e, para cada atividade de mais baixa prioridade, utilizar a sua folga livre e depois a folga total.

Boiteux [02] avalia que nem sempre é possível remanejar todas as atividades, otimizando o uso de todos os recursos, mas, deve-se ter em mente que a menor otimização sempre é preferível a nenhuma otimização. O uso dos recursos do exemplo de projeto acima, poderia ser remodelado, fazendo-se as seguinte modificações:

Atividade	Depende de:	Duração (semanas)	+ cedo		+ tarde		Folga	Cronograma											
			Ini	Fim	Ini	Fim		1	2	3	4	5	6	7	8	9	10		
BD1	-	3	1	3	2	4	1	█	█	█	█								
BD2	-	4	1	4	1	4	-	█	█	█	█								
PGM 1	BD1	2	4	5	9	10	5				█	█	█	█	█	█	█	█	█
PGM 2	BD 1 e 2	6	5	10	5	10	-					█	█	█	█	█	█	█	█
PGM 3	BD 2	4	5	8	7	10	2											█	█
Recursos	Administrador de dados							2	2	2	1								
	Programador											2	2	2	2	2	2	2	2
	Computador							2	2	2	1	2	2	2	2	2	2	2	2



A atividade PGM 1, foi deslocada para o final da sua folga livre, isto é 9^a e 10^a semanas, provocando assim, um alívio na carga de recursos da 4^a e 5^a semanas.

7. O problema da aceleração

A questão da aceleração consiste em procurar condições no projeto que propiciem a redução do tempo, garantindo-se a entrega das mesmas funcionalidades.

Para que isto seja possível, Boiteux [02] esclarece que deve-se estimar os tempos necessários para cada uma das atividades, considerando, no mínimo duas situações:

- A situação normal de execução da tarefa;
- A situação mais otimista em que se pode realizar a tarefa, garantindo o mesmo padrão de qualidade.

Em termos práticos, a questão se coloca em efetuar os cálculos dos tempos citados no discussão do custo direto:

- t_A = tempo de desenvolvimento de forma acelerada.
- t_N = tempo normal (ótimo) de desenvolvimento do sistema.
- C_A = custo correspondente ao tempo de desenvolvimento de forma acelerada
- C_N = custo correspondente ao tempo normal (ótimo) de desenvolvimento do sistema

Para o acompanhamento do cálculo da aceleração, considerar-se-á um projeto, tal como o que segue, formado por oito atividades:

<i>Projeto</i>		
<i>Atividade</i>	<i>Descrição</i>	<i>Depende de:</i>
A	Modelagem conceitual	-
B	Elaboração do componente 1	A
C	Elaboração do plano de entrega	-
D	Elaboração do Componente 2	A
E	Elaboração do programa 1	B
F	Elaboração do programa 2	B
G	Elaboração do programa 3	E, F
H	Testes finais e entrega	C, F, G

Para esse projeto foram, primeiramente, estimadas três estimativas de duração para cada uma das atividades:

- a mais otimista - mínima,
- a mais pessimista – máxima e

- a esperada, bem como foram estimados os seus respectivos valores de custo, considerando as alternativas mínima e esperada. Evidentemente, os custos referentes à alternativa mínima são maiores, pois representam o sacrifício necessário para a devida redução do tempo. Esse adicional de custo, representa, por exemplo, horas adicionais pagas aos desenvolvedores.

Os cálculos da rede PERT, considerando a estimativa esperada, são os seguintes:

Projeto									
Atividade	Estimativa de Duração			Depende de:	+ cedo		+ tarde		Folga
	+ pessimista	+ otimista	Esperada		Início	Fim	Início	Fim	
A	3	2	3	-	1	3	1	3	-
B	4	3	4	A	4	7	4	7	-
C	5	4	6	-	1	6	16	21	15
D	3	2	4	A	4	7	14	17	10
E	6	5	6	B	8	13	12	17	4
F	12	9	14	B	8	21	8	21	-
G	4	2	4	D, E	14	17	18	21	4
H	3	2	3	C, F, G	22	24	22	24	-

O caminho crítico é formado pelas atividades: **A, B, F e H**

Considerou-se que o montante de custos indiretos corresponde, inicialmente a \$ 2.160.

Considerando o exposto para o comportamento dos custos diretos, calculou-se a relação entre a variação de custo e a variação do tempo, que mede o custo unitário ou o custo marginal de aceleração:

Atividade	Duração esperada ()		Duração mínima		Cálculo do custo marginal de aceleração		
	Duração (t_N)	Custo (\$) (C_N)	Duração (t_A)	Custo (\$) (C_A)	Varição do tempo	Varição do Custo	Custo marginal
A	3	120	2	180	1	60	60
B	4	160	3	210	1	50	50
C	6	1.000	4	1.200	2	200	100
D	4	160	2	240	2	80	40
E	6	300	5	330	1	30	30
F	14	560	9	960	5	400	80
G	4	160	2	280	2	120	60
H	3	120	2	190	1	70	70
Custo Total		2.580		3.590			

O cálculo da aceleração do projeto é feito através da construção de uma matriz, onde o conceito básico é a redução gradual dos tempos de execução de cada uma das atividades críticas (cada vez, reduz-se apenas uma unidade de tempo), até se chegar ao tempo mínimo necessário para cada uma delas.

A redução de tempo inicia-se pela atividade que apresentar o menor custo marginal de aceleração. A nova composição de tempos, caminho crítico e custo direto se encerra somente quando foram reduzidos todos os tempos das

atividades críticas. Este então representa o maior esforço possível de aceleração do projeto. Tentativas de reduzir mais o tempo, implicarão em entregas do produto com menos funcionalidades.

A cada redução de tempo, soma-se o custo marginal de aceleração ao custo direto total previsto. Espera-se, entretanto, que haja uma redução no custo indireto, correspondente àquela porção que o projeto deixa de absorver, pelo fato de o seu tempo estar sendo reduzido. No exemplo, este valor corresponde a uma participação de \$ 90 por unidade de tempo.

O projeto poderia ser realizado na faixa de tempo correspondente ao tempo esperado, em situações normais e o tempo mais otimista.

Resumidamente, a matriz abaixo demonstra os cálculos necessários, passo a passo até se obter a nova composição de tarefas e custo:

Projeto											
	Atividades								Custos		
	A	B	C	D	E	F	G	H	Direto	Indireto	Total
R ->	60	50	100	40	30	80	60	70			
Prioridade de redução ->	2º	1º	-	-	-	4º	-	3º			
Duração ↓											
24	3	4 ↓	6	4	6	14	4	3	2.580	2.160	4.740
23	3 ↓	3 ↓	6	4	6	14	4	3	2.630	2.070	4.700
22	2	3	6	4	6	14	4	3 ↓	2.690	1.980	4.670
21	2	3	6	4	6	14	4	2 ↓	2.760	1.890	4.650
20	2	3	6	4	6	13 ↓	4	2	2.840	1.800	4.640
19	2	3	6	4	6	12 ↓	4	2	2.920	1.710	4.630
18	2	3	6	4	6	11 ↓	4	2	3.000	1.620	4.620
17	2	3	6	4	6	10 ↓	4	2	3.080	1.530	4.610
16	2	3	6	4	6	9 ↓	4	2	3.190	1.440	4.630

Somente após a composição final do diagrama correspondente ao cronograma físico-financeiro, que corresponde ao tempo mais otimista para a execução do projeto, é possível alocar convenientemente os recursos necessários. Qualquer alocação feita anteriormente a este cálculo, não tem sentido algum, pois certamente será modificado.

8. Conclusões

Da mesma forma que pode ser calculado uma aceleração do projeto, aplicando-se a redução gradual sobre as atividades consideradas críticas, partindo-se do conceito que o custo direto cresce quando o tempo diminui e vice-versa, pode-se diminuir o custo de um projeto sistêmico, aumentando o seu tempo de execução.

O modelo apresentado é uma contribuição ao complexo ambiente em que ocorre o gerenciamento do desenvolvimento de software. Além dos benefícios imediatos da aplicação desses conceitos, apresentação dos resultados e as possibilidades de negociação do gerente ganham um importante aliado.

Acredita-se que apresentar os resultados da atividade de desenvolvimento de sistemas em termos monetários é uma das poucas formas eficazes de se sensibilizar os executivos dessas empresas, quanto a importância na condução dos negócios de Informática.

GLOSSÁRIO

CMM – Capability Maturity Model - é uma iniciativa do SEI (Software Engineering Institute) para avaliar e melhorar a capacitação de empresas que produzem software; o projeto CMM foi apoiado pelo Departamento de Defesa do Governo dos Estados Unidos; embora não seja uma norma emitida por uma instituição internacional (como a ISO ou o IEEE), tem tido uma grande aceitação mundial, até mesmo fora do mercado americano. O CMM é um modelo para medição da maturidade de uma organização no que diz respeito ao processo de desenvolvimento de software, classificando as organizações em cinco níveis distintos, cada um com suas características próprias: inicial, repetível, definido, gerenciado e otimizado. O CMMI - Capability Maturity Model Integrated , é a nova versão do CMM, envolvendo as características de adequação aos objetivos de cada empresa.

SPICE - Software Process Improvement and Capability dEtermination - ISO 15504 - é uma norma em elaboração conjunta pela ISO e pelo IEC. Ela constitui-se de um padrão para a avaliação do processo de software, visando determinar a capacitação de uma organização, visando ainda orientar a organização para uma melhoria contínua do processo.

PMI – Project Management Institute – É um organismo de estudo em administração de projetos, que procura divulgar a prática das melhores práticas em gerenciamento de projetos. O documento onde consta essas práticas é o PMBOK – Guide to the Project Management Body of Knowledge.

REFERENCIAS BIBLIOGRÁFICAS

- [01] ARTHUR, Lowell Jay, *Produtividade do programador*, Rio de Janeiro: Editora Campus, 1992, 1^a. edição.
- [02] BOITEUX, Colbert Demaria, *Pert / Com / Roy e outras técnicas de programação e controle*, Rio de Janeiro: LTC, 1985.
- [03] DEMARCO, Tom, *Controlling software projects*, New-York, Prentice-Hall, Inc, 1982.
- [04] FERNANDES, Aguinaldo Aragon, *Gerência de software através de métricas - garantindo a qualidade do projeto, processo e produto*, São Paulo: Editora Atlas, 1995, 1^a. edição.
- [05] GARRISON, Ray H., *Managerial accounting concepts for planning, control and decision making*, Dallas: Business Publications, 1976.
- [06] HEHN, Herman Frederico, *Peopleware - como trabalhar o fator humano nas implementações de sistemas integrados de informação*, São Paulo: Gente, 1999, 1^a.edição.
- [07] LEONÉ, George Sebastião Guerra, *Custos - planejamento, implantação e controle*, São Paulo: Editora Atlas, 1996, 2^a. edição.
- [08] NAKAJIMA, Seiichi, *Introdução ao TPM – total productive maintenance*, São Paulo: IMC Internacional Sistemas Educativos, 1989.
- [09] PAGE-JONES, Meillir, *Gerenciamento de projetos - uma abordagem prática e estratégica*, São Paulo: Makron Books, 1990, 2^a. edição.

- [10] PRESSMAN, Roger S., *Software Engineering*, New-York: McGrawHill, Inc, 1999 – 5ª edição.
- [11] VALERIANO, Dalton L., *Gerência em projetos - pesquisa, desenvolvimento e engenharia*, São Paulo: Makron Books, 1998.