

GESTÃO DE CADEIAS CRÍTICAS DE MÚLTIPLOS PROJETOS SIMULTÂNEOS

1 - Introdução

Muitos projetos experimentam falhas sérias em termos de prazo, orçamento e âmbito, e a taxa de projetos cancelados e/ou mal sucedidos têm-se mantidos a níveis alarmantes. A gestão de cadeias críticas de projetos consiste num método que permite concluir projetos em tempos significativamente mais curtos do que com a gestão tradicional baseada no caminho crítico. É de uso mais simples e requer menos trabalho à equipe de projeto. [Goldratt, 1992]

O método da Cadeia Crítica consiste primeiramente em conseguir estimativas dos tempos de processamento reais, em seguida resolver os problemas de restrições de recursos através das precedências extras, após o qual se identifica a cadeia crítica, que tem em linha de conta as restrições de recursos. Como os tempos de processamento não têm buffers, é inserido um buffer para todo o projeto, em vez de cada atividade ter o seu próprio buffer, construído com base na cadeia crítica. As cadeias que alimentam a cadeia crítica devem ter também um buffer para não influenciarem a mesma. Dimensionam-se os buffers, para ter uma boa segurança, por exemplo, com 50% do tempo de processamento das atividades que seguram, e calendariza-se o mais tarde possível. Ao realizar as atividades o mais tarde possível, haverá um maior conhecimento do projeto quando a atividade se realizar, e também se pagará esta o mais tarde possível sem comprometer a duração total do projeto. Quando uma cadeia começa, as atividades devem ser feitas o mais rápido possível (ASAP), devendo a atividade seguinte estar pronta a ser realizada logo que a atividade precedente acabar.

2 - Objetivos

Estudar as aplicações práticas da metodologia de cadeias críticas com múltiplos projetos simultâneos, estruturado na utilização e comparação das três aproximações apresentadas por Newbold (1998) : Aproximação de todos juntos, aproximação de projetos sucessivos e aproximação de recurso estratégico ou gargalo.

3 - Análises e Conclusões

3.1 - Metodologia

Diferentes variações de um problema exemplo foram desenvolvidas com a finalidade de provocar situações de certo grau de complexidade durante a programação-base e execução dos projetos que permitiram analisar as vantagens e deficiências existentes ao aplicar a metodologia mediante a ferramenta de software de gestão de projetos ProChain, do Microsoft Project. Similarmente, procurou-se identificar a existência de discrepâncias entre a teoria e prática da metodologia. Assim analisou-se o desempenho que têm os diferentes *buffers* de projeto em situações de atraso e dependência entre múltiplos projetos.

3.2 - Aproximações

- A primeira aproximação de **“Todos juntos”** toma todas as atividades de todos os projetos e as programa simultaneamente, dentro de um único projeto criado artificialmente. Esse projeto artificial terá uma data final que representará o término de todos os projetos e será programado como se fosse um único projeto individual. Isto é, no qual serão incluídos *“buffers de alimentação”* - *BA* e *“buffers de projeto”* - *BP*, levando em conta a totalidade de atividades dos três projetos. Não são utilizados *“buffers tambor”* (ou *buffers do recurso gargalo*), pois não se faz diferença explícita entre os projetos. Além disso, se espera que o BA cumpra a função de proteção do recurso estratégico.

Programam-se as atividades com as datas-mais-tarde, levando em conta todas as restrições de datas de entrega. Assim, se pode identificar a seqüência de

atividades mais longa que não tenha folga nem para o futuro nem para o passado. Essa seria a cadeia crítica do projeto artificial.

- A segunda aproximação de “**Projetos sucessivos**” parte da programação (*Schedule*) individual de cada projeto. O nivelamento, inserção de *buffers* e identificação da cadeia crítica nessa programação prévia são feitos supondo a disponibilidade total de todos os recursos, como se o projeto fosse único. Uma vez resolvidos os conflitos de recursos em cada projeto pode-se continuar com o programa de múltiplos projetos ou *Máster*. Neste, se inserem os projetos individuais organizadamente seguindo uma escala prioritária.

Nessa aproximação a carga de trabalho é nivelada em todos os recursos mediante o encaixamento de cada projeto que forma parte do *Máster*. Ou seja, se for necessário, serão alteradas as datas de início dos diferentes projetos, atrasando o início do projeto de menor prioridade até que não exista mais algum conflito de disponibilidade de recursos. O normal é que cada novo projeto no sistema seja colocado no final e com uma prioridade menor. Se, eventualmente, tiver espaços livres entre os projetos já programados, pode-se antecipar a programação de atividades do projeto novo, mas, tendo como limite que a última atividade não esteja antes da data estabelecida.

Esse tipo de programação permite lidar de forma mais eficiente com a organização de cada projeto. Entretanto, à medida que o número de projetos simultâneos aumenta e o grau de complexidade, em função das dependências de recursos, se eleva, a eficiência irá se deteriorando, podendo ser impossível atingir o nivelamento de todos os recursos de forma adequada.

- A terceira aproximação de “**Recurso estratégico ou gargalo**” tem a mesma concepção da aproximação de “projetos sucessivos”, utilizando, também, o programa especial “*Máster*”. Todavia, esta última se diferencia na etapa de nivelamento de carga de todos os projetos, resolvendo os conflitos de recursos entre projetos unicamente para o recurso definido como gargalo do sistema. Supõe-se que só o recurso estratégico será gargalo para o sistema e que, ao resolver os conflitos do gargalo, serão criados espaços que permitam solucionar conflitos nos outros recursos.

Certamente é de se esperar a aparição de novos conflitos em recursos não-gargalos e procedimentos alternativos de gestão e programação para mitigar o efeito desses conflitos serão necessários.

3.3 - Os casos estudados e conclusões

1 - Programação Normal:

Utilizam-se os parâmetros do software estabelecidos por omissão (*default*), considera-se que desde o início da programação básica se conhece com certeza a chegada dos três projetos em diferentes datas, vale dizer, o programa contém os três projetos todo o tempo. Respeita-se a prioridade original, considerando que o projeto mais prioritário é o primeiro em carteira. O grau de importância diminui de acordo com a ordem de chegada.

Neste caso e na maioria dos outros casos, utilizamos os parâmetros de software estabelecidos por omissão, utilizando dois tipos de parâmetros distintos. Os da ferramenta Prochain, utilizada na primeira aproximação e na parte inicial (programação individual) das segunda e terceira aproximação. E os da ferramenta Pipeline utilizados para criar o programa global ou *Máster* nas duas últimas aproximações. Vale a pena lembrar que para utilizar o Pipeline é preciso inserir os projetos previamente programados no Prochain.

No caso dos parâmetros de Prochain nos referimos às opções de como fazer os principais passos de programação da metodologia da cadeia crítica num projeto individual.

1. Nivelamento:
 - a. Otimizando ou não (*sim*)
2. Identificação da cadeia crítica:
 - a. O caminho mais longo (*sim*)
 - b. O caminho mais restrito
3. Criação de *buffers*:
 - a. Método das porcentagens (*sim, 50%*)
 - b. Método da soma de quadrados

4. Inserção de *buffers*:
 - a. Consolidar o risco e/o (*sim*)
 - b. Criar o programa de linha de base (*sim*)
5. Acompanhamento:
 - a. Aplicar os tempos atuais nos relacionamentos entre atividades (*sim*)
 - b. Criar um quadro de estado dos *buffers* (*sim*)
 - c. Otimizar os tempos projetados

As opções marcadas com (*sim*) são as utilizadas por omissão. O uso desses parâmetros faz com que a metodologia seja aplicada de acordo com a teoria básica encontrada na literatura, à exceção da opção de *consolidar o risco* que foi estabelecida como parâmetro por omissão pela compactação visual obtida no programa, pois sua aplicação não afeta a data final do projeto. Os outros parâmetros não causam uma diferença relevante no problema estudado, mas é recomendado usá-los para a programação de projetos complexos.

2 - Programação Prioritária:

Difere do anterior no ordenamento prioritário dos projetos. Neste caso, a prioridade arbitrariamente estabelecida é preemptiva em relação à prioridade de ordem de chegada. Igualmente trabalha-se o tempo todo com os três projetos.

3 - Programação de Inserção:

Difere dos anteriores por considerar que inicialmente só se tem conhecimento da existência de dois projetos e que, numa determinada data posterior, é preciso inserir um projeto no programa global. Este último projeto tem uma prioridade menor e é programado por último. Neste caso também se analisará as características do programa ao inserir projetos definidos com maior prioridade dos que estão em andamento. Duas variantes podem ser utilizadas neste cenário.

Quando a chegada do novo projeto é tratada com uma *reprogramação parcial*. - Mantêm-se fixo o estado dos programas em execução até o momento de inserir o projeto novo. Nesse momento unicamente serão programadas as atividades do último projeto a fim de encaixá-lo no programa global *Máster*.

Quando a chegada do novo projeto é tratada com uma *reprogramação total*. - No momento da inserção se realiza uma reprogramação geral, onde segundo as prioridades estabelecidas se recalculam a estrutura da rede, tamanho de *buffers*, as datas de início e fim de todos os projetos.

Todos os casos são desenvolvidos nas duas últimas aproximações (*Projetos sucessivos e O recurso estratégico ou gargalo*) e só alguns na primeira aproximação (*Todos juntos*). Isso porque a aproximação, pelo fato de ser simples e ter um tratamento de projeto individual, não permite variações do tipo prioritário.

Em todos os casos provocam-se diversos atrasos de atividades na etapa de execução. Embora estes distúrbios não sejam exatamente iguais entre os casos, procuram ter o mesmo efeito e ser coerentes com a lógica do programa (*Schedule*).

Referências

A branch-and-bound procedure for the multiple resource-constrained project scheduling problem

Demeulemeester, Erik; Herroelen, Willy.

Management Science; Dec. 1992; 38, 12; ABI/INFORM Global
pg. 1803

International Journal of Project Management 18 (2000) 173±177

Critical chain: the theory of constraints applied to project management

Graham K. Rand*

Critical chain project scheduling: Do not oversimplify

Willy Herroelen; Roel Leus; Erik Demeulemeester

Project Management Journal; Dec. 2002; 33, 4; ABI/INFORM Global
pg. 48

Journal of Operations Management 19 (2001) 559–577

On the merits and pitfalls of critical chain scheduling

Willy Herroelen*, Roel Leus