

# Os Problemas em Cronogramas

## *Decorrentes das Falhas no CPM*

Leo Sumner e Mat Barnes

Depois de décadas criando e analisando o CPM (Método do Caminho Crítico) dos cronogramas e fazendo melhoria sobre o controle de projetos através de práticas para o aperfeiçoamento de cronogramas, temos uma grande preocupação ao constatar um aumento do CPM nos cronogramas preparados para os grandes projetos, contendo numerosas e diversas falhas e uso indevido. ☹



**Leo Sumner** é o sócio-fundador do Tiburon Group, LLC (TTG) e atua como diretor responsável por inúmeros projetos. Ele também opera em diversos projetos como gerente e programador de Controle do Projeto. Adicionalmente, ele comanda o Serviço de Atendimento a Reclamações TTG e atuou como testemunha especialista. Ele tem mais de 30 anos de experiência em todas as fases de design, construção e gerenciamento de projetos. Ele trabalhou com projetos educacionais, de tratamento de saúde, de bibliotecas, laboratórios, trânsito, justiça criminal, geração de energia, petroquímicas e projetos geotérmicos. Ele tem um BS em Tecnologia Industrial pela Universidade Southern Illinois, Carbondale IL e um MBA do St. Mary's College, Moraga CA.



**Mat Barnes** é diretor na TTG e atua como programador sênior e engenheiro de Reclamações em seu Departamento de Controle de Projetos. Ele tem mais de 25 anos de experiência na indústria da construção trabalhando em prédios e projetos de construção pesada em âmbito mundial. No decorrer dos anos, suas responsabilidades passaram de engenheiro de projetos, programador e engenheiro residente para gerente de projetos e de construção. Mat realiza programações, revisões de cronogramas e serviços de atendimento a reclamações para clientes da TTG e atuou como testemunha especialista em reclamações de construção arquivadas. Ele realizou estes serviços em projetos abrangendo desde escolas K-12, hospitais, projetos de grandes infra-estruturas, minerações, usinas de geração de energia nuclear e projetos da marinha. Ele é um especialista no Método do Caminho Crítico para Programação e é especialista no uso de uma variedade de pacotes de softwares para programação incluindo o Primavera P3, SureTrak, Promis, Fast Track e MS Project.

Poucos profissionais da área da construção afirmariam que o CPM de um cronograma bem desenvolvido num projeto importante é um marco para se ter um bom sistema de controle de projetos implantado. Ele deveria ser uma das ferramentas mais importantes para todos os stakeholders do projeto. No entanto, se o CPM de um cronograma contiver falhas e uma utilização fundamental inadequada, ele se torna uma ferramenta amplamente inútil.

Temos experiência de que a maioria dos proprietários envolvidos nos principais projetos de construção estabelece imparcialmente sofisticados requisitos nos cronogramas em seus projetos. As especificações irão detalhar o uso do CPM, limitar a duração das atividades, direcionar o percentual de atividades críticas e quase críticas, demandar as cargas de custos e de recursos e podem incluir uma variedade de outros requisitos num esforço a fim de assegurar que controles adequados de projeto sejam aplicados em seus projetos. Eles têm o direito de requerer isto, uma vez que eles estão pagando pelo projeto. Alguém poderia pensar que os contratantes e seus subcontratantes compartilhariam este desejo. Deve-se afirmar que temos experiência que a maior parte dos contratantes o faz.

### Os “as built” e as disputas na construção

Em nossa indústria atual, os CPMs dos cronogramas “as built” (conforme realizado) transformaram-se na ferramenta primária para julgar disputas referentes a atrasos e alterações. Uma vez que o cronograma “as built” é o resultado da atualização precisa da baseline (linha base) numa base periódica regular e a baseline deve “presumivelmente” ser acatada por força contratual e ser bem desenvolvida, alguém pode se perguntar como tantos cronogramas aparentemente vistosos se transformam num pântano que mais se parece com um prato de macarrão do que com um cronograma as built de um projeto. Nós acreditamos que uma boa parte disto se deve a falhas no uso dos bons fundamentos do CPM no início do projeto durante o desenvolvimento do cronograma e à falta de habilidade dos programadores de prazos/gerente de projeto para identificar e corrigir os problemas que surgem daí. Isto, associado à tendência natural de se transferir a responsabilidade sobre os atrasos do projeto para os outros, **resulta num cronograma as built que é inadequado para o planejamento do projeto e de valor questionável para análises legais.**

Poucos anos atrás houve um artigo bastante interessante sobre os Registros da Nova Engenharia que parece ter passado despercebido pela maioria dos stakeholders de nossa indústria, mas que deu uma agitada dentre os profissionais da comunidade da programação de prazos. Para ser mais específico, o artigo era de 26 de maio de 2003, editado por Richard Korman e Stephen H. Daniels. Ele chamava-se *“Os críticos não acham a lógica em muitos dos CPM em cronogramas atuais – Os usuários desejam um software com flexibilidade, mas é um CPM real?”*.

### Falhas no projeto do software para programação ou na aplicação?

O tema central daquele artigo é, por assim se dizer, como os computadores transformaram a programação de prazos CPM e ele inclui ainda um debate entre quatro especialistas sobre as falhas que eles vêem no emprego de bons fundamentos da programação CPM e o papel destas falhas na redução da qualidade dos cronogramas usualmente preparados hoje em dia. Há, naquele artigo, uma indicação apontando na direção do software Inc.’s P3 da Primavera Systems, já que eles possuem a parte grande do mercado de softwares para programação de prazos. A crítica básica é a de que a Primavera desenvolveu um software que é flexível demais e, para não dizer, muito poderoso, que possibilita mau uso da programação de prazos e, por falta de observação dos fundamentos do CPM, gera sérias falhas ao cronograma.

Richard K. Faris, um dos fundadores da Primavera, rebate no artigo afirmando que as críticas queriam que a Primavera incorporasse funções em seu software que fariam as pessoas utilizarem o programa de uma forma específica. Elas também afirmavam, e acreditamos, de forma correta, que os usuários de seu software não o comprariam se ele não fosse potente e flexível. Mais especificamente, ele afirmou que *“a implicação está completamente errada que a Primavera pode controlar o modo como seu software é utilizado. A empresa fornece um produto robusto e versátil focado nas necessidades de seus usuários. A Primavera não pode ser responsabilizada por mau uso da mesma forma que um desenvolvedor de planilhas não é responsável por aqueles que utilizam seu produto para gerar relatórios falhos ou enganosos.”* E a gente tem que concordar com isto.

Então, qual é o problema de fato? Quem tinha razão no debate? Os quatro especialistas em programação, a Primavera, ou ambos? Não é realístico esperar que uma ferramenta de con-

trole de projetos seja projetada para servir de “mordça”. Tampouco é realístico que a Primavera, ou qualquer outra empresa de software, seja responsabilizada pelo mau uso do software por terceiros. Nós também insistimos que os especialistas em programação têm razão em suas críticas.

O software no mercado atual é relativamente barato e muito fácil de assimilar. Isto pode levar à tentação de se ter pessoas inexperientes fazendo programação. Se pessoas inexperientes estiverem desenvolvendo o cronograma, os fundamentos do CPM podem ser ignorados. Para citar o artigo.... “A programação afastou-se de seus papas na direção do homem comum e há poucas pessoas capazes fazendo isto,” afirma Faris. Ele diz que **treinar ainda é vital**. Treinar, deve ser ressaltado, **nos fundamentos do Método do Caminho Crítico de programação e não no emprego do software.**

A base é a de que as pessoas que desenvolvem CPM nos cronogramas são fortemente treinadas somente no uso do software e não estão familiarizadas com os fundamentos do CPM. Ampliando o problema, existe a questão que uma quantidade considerável de pessoas que examinam os cronogramas para os proprietários também não são treinadas sobre os fundamentos do CPM, nem em análise de cronogramas. Até mesmo cronogramas apropriadamente estruturados desenvolvidos para nossos projetos podem ser muito complexos e de difícil compreensão. Eles podem conter inúmeras características que, de fato, fazem o que “presumivelmente” é um CPM se tornar em um não-CPM. Isto pode ser intencional ou não. Independentemente da intenção, o resultado final é uma ferramenta que é mal empregada e inútil.

### Premissas básicas da programação CPM

Não é nossa intenção aborrecer o leitor com uma dissertação sobre a programação CPM. Contudo, é importante falar sobre as premissas básicas do CPM. Os fundamentos básicos do CPM utilizam tanto conceitos sobre o método de diagramação de flexas (ADM – Arrows Diagramming Method) quanto sobre o método de diagramação de precedências (PDM - Precedence Diagramming Method). Por ser menos flexível do que o PDM, o ADM tem a vantagem de definir as relações lógicas entre as atividades inteiramente através dos números das atividades. O PDM é uma técnica mais flexível que evita usar atividades suspeitas a fim de manter as relações lógicas necessárias do ADM. O PDM representa as atividades como caixas e atribui as propriedades das atividades a estas caixas. Isto inclui os quatro tipos de relações de intervalos: término-início, início-início, término-término e início-término. Esta flexibilidade é uma das causas principais dos cronogramas falhos. Com exceção da lógica término-início, nos referimos às outras relações como “não convencionais” e o problema começa quando elas não são utilizadas ou coordenadas adequadamente. O problema pode ser agravado quando são feitas alterações unilaterais numa operação de atualização do

cronograma, quando a lógica é revisada e não manipulada apropriadamente.

Numa ação judicial recente, fomos solicitados a revisar o cronograma e suas várias atualizações mensais feitas por um contratante. Não nos foi solicitado posicionarmos frente a nenhuma causa específica dos atrasos, mas simplesmente de comentar sobre a qualidade dos cronogramas e sobre sua utilidade como ferramenta de planejamento e de previsão. Nós encontramos uma variedade de falhas estruturais na baseline em questão, a maioria delas bastante comum que, quando corrigidas, **previam o término do projeto três semanas depois da data de término contratual**. É importante ter em mente que este era o cronograma do projeto, que representava o plano de construção do contratante de como seria antes do início dos trabalhos. Infelizmente, nenhum dos stakeholders do projeto – nem o proprietário, nem o arquiteto, nem o contratante geral ou seus subcontratados – sabia disto e, além do mais, eles confiavam no cronograma para fins de gerenciamento e de previsões. Compondo a questão, ao preparar a primeira atualização mensal do cronograma, o contratante, evidentemente reagindo a atrasos na obra já na fase inicial, procurou melhorar o cronograma revisando sua lógica. Agindo assim, ele negligenciou a lógica de relacionamento anterior (término-início, início-início, término-término e início-término) para adequar seus prazos conforme as datas contratuais. Ajustar um cronograma simplesmente alterando a lógica de relacionamento num cronograma com centenas de nós de relacionamento, é um erro. A consequência deste erro simples foi que, no sétimo mês do projeto, as atualizações do contratante lhe mostraram, a duas semanas da data de término do contratante, que ele estava, de fato, bem mais de cem dias atrasado em relação ao cronograma e nem ele, nem o proprietário tinham consciência disto.

### Falhas comuns em cronogramas

Muitas coisas que podem reduzir a utilidade do cronograma que você está empregando em seus projetos, todas elas perfeitamente admissíveis em termos de software de programação, mas não em termos de boas práticas de CPM. Destas, algumas das mais comuns são:

- **Uso excessivo ou inadequado de lógica não-convencional.** Quer dizer, qualquer lógica diferente de término-início. Lembre-se, o ADM tem somente um nó de lógica: término-início. O PDM aceita os demais. Quando utilizados adequadamente, eles são válidos e podem ser bastante úteis. Empregados inadequadamente, eles podem sujeitar o cronograma a falhas e demandar reparos. Além disso, o emprego excessivo de lógica não convencional pode gerar uma situação na qual ocorrem erros. Resumidamente, se alguém emprega relações início-início excessivamente, aumenta a probabilidade de falhas por não considerar na coordenação a relação término. Ou nas atualizações, aumenta a probabilidade de falha de coordenação por não voltar à lógica convencional de relacionamento na revisão do cronograma. Também poderá causar