

MODELOS PARA GERAÇÃO DE EVENTOS EM ANÁLISE DE CUSTO DE VIDA ÚTIL DE SISTEMAS TÉCNICOS

Eduardo Siqueira Brick

Universidade Federal Fluminense (UFF) – Núcleo de Logística Integrada e Sistemas (LOGIS)
Rua Passo da Pátria nº 156, sala 440, Bloco E, S. Domingos, Niterói - RJ
brick@producao.uff.br

Denis Gonçalves Cople

Universidade Federal Fluminense (UFF) – Núcleo de Logística Integrada e Sistemas (LOGIS)
Rua Passo da Pátria nº 156, sala 440, Bloco E, S. Domingos, Niterói - RJ
deniscople@terra.com.br

Resumo

Com a crescente complexidade dos sistemas técnicos, torna-se necessária uma pré-avaliação de sua implementação, operacionalização e desativação sob a ótica do custo, o que é chamado de Análise de Custo de Vida Útil (ACVU).

A metodologia de ACVU envolve a identificação de elementos de custo e a sua alocação ao longo do tempo. Cada um desses elementos pode ocorrer em instantes distintos ao longo do ciclo de vida do sistema e pode empregar uma fórmula de cálculo específica.

Os elementos de custo devem ser organizados hierarquicamente, com o uso de uma estrutura em árvore denominada Estrutura de Desdobramento de Custos (EDC).

Um dos problemas mais importantes na construção de uma ferramenta que possibilite efetivar o cálculo do custo de vida útil de um sistema é a identificação dos instantes em que ocorrem os eventos associados aos átomos de custo definidos pela EDC, a fim de possibilitar a aplicação de taxas de desconto, referenciando todos os custos a uma mesma data.

Dentro deste contexto, o propósito principal deste trabalho foi o de descrever uma abordagem que pode ser usada para criar modelos de geração de eventos, apropriados para uso em uma ferramenta computacional para ACVU.

Palavras-Chaves: Engenharia de Sistemas, Apoio Logístico Integrado, Simulação, Análise de Custo de Vida Útil

Abstract

The growing complexity of technological systems has increased the need for an “a priori” assessment of its implementation, operation and deactivation costs, considering all phases of its life cycle, using a type of analysis known as Life Cycle Cost Analysis (LCCA).

The LCCA methodology involves the identification of cost elements and its allocation to specific time frames. Each one of these cost elements may occur at distinct instants along the system life cycle and use specific calculation formulas.

The cost elements must be organized in a tree like cost structure called CBS – Cost Breakdown Structure.

One of the most important problems for the construction of a tool to perform LCCA is the identification of the instants of occurrence of the events associated with the cost atoms defined by the CBS, in order to make it possible to apply discount techniques, translating all costs to the same date.

In this context, the goal of this paper was to describe an approach that may be used to create event generation models, appropriated for the implementation of LCCA software tools.

Keywords: Systems Engineering, Integrated Logistic Support, Simulation, Life Cycle Cost Analysis

1. INTRODUÇÃO

Com a crescente complexidade dos sistemas técnicos, torna-se necessária uma pré-avaliação da sua implementação, operacionalização e desativação, sob a ótica do custo, o que é denominado Análise de Custo de Vida Útil (ACVU).

As questões ligadas à ACVU são as mais diversas, envolvendo desde o estudo da confiabilidade e da manutenibilidade de equipamentos, até a análise das conseqüências de falhas, ou do pessoal necessário à operação do sistema. Muitos são os modelos que podem ser utilizados na análise, mas todos com um mesmo fim, que é estimar o custo total do sistema durante toda a sua existência.

Segundo Fuller&Petersen (1996), a ACVU pode ser definida como uma metodologia de avaliação econômica de projetos, na qual são considerados todos os custos relativos à aquisição, desenvolvimento, operação, manutenção e desativação, desde que os mesmos sejam considerados potencialmente importantes para uma tomada de decisão. Esta metodologia se mostra particularmente interessante quando devem ser testadas diferentes alternativas de projeto que satisfazem a um dado planejamento de construção (incluindo elementos como conforto, segurança, padronização de componentes, confiabilidade, manutenibilidade, disponibilidade, desempenho e qualquer consideração estética), mas que podem ter diferentes investimentos iniciais; custos diferentes para operação, manutenção e reparo; gastos com energia e, possivelmente, diferentes tempos de vida útil.

A metodologia de ACVU envolve a identificação de elementos de custo e a sua alocação ao longo do tempo. Cada um desses elementos (ou átomos de custo) pode ocorrer em instantes distintos ao longo do ciclo de vida do sistema (denominados eventos) e tem uma fórmula de cálculo específica.

Os elementos de custo são usualmente organizados em uma estrutura hierárquica arborescente, denominada Estrutura de Desdobramento de Custos (EDC).

Um dos problemas mais importantes na construção de uma ferramenta que possibilite efetivar o cálculo do custo de vida útil de um sistema é a identificação dos instantes em que ocorrem os eventos associados aos átomos de custo definidos pela EDC.

O problema, na sua forma mais geral, embora conceitualmente simples, pode se tornar computacionalmente bem complexo, quando se considera que muitas unidades (instâncias) de um sistema podem ser produzidas e operadas em locais, períodos de tempo e formas distintas. Além do mais, muitos dos eventos relevantes dependem dos estados possíveis que o sistema técnico pode ocupar durante a sua vida e a transição entre estes estados, por sua vez, depende da ocorrência de outros eventos. Existe, assim, uma dependência circular entre alguns eventos.

Dentro deste contexto, o propósito deste trabalho foi o de descrever algumas abordagens que podem ser usadas para criar modelos de geração de eventos apropriados para implementação de uma ferramenta computacional para ACVU.

2. ATIVIDADES E ESTRUTURAS DE DESDOBRAMENTO DE CUSTOS

Segundo Pilla (2003) a identificação adequada de todos os custos a serem incorridos ao longo do ciclo de vida de um sistema depende fundamentalmente do desenvolvimento de uma Estrutura de Desdobramento de Custos. Esta consiste numa espécie de “lista” ou “plano de contas”, geralmente organizado de forma hierárquica, que apresenta todos os itens de custo a serem considerados no processo de ACVU.

Uma EDC deve apresentar, pelo menos, as seguintes características (BLANCHARD, 1998):

- todos os elementos de custo do sistema devem ser considerados;
- cada categoria de custo é relacionada a uma função, um nível de atividade e/ou algum item de *hardware/software*;
- a estrutura deve ser codificada de maneira hierárquica e as categorias devem ser organizadas de modo a permitir a análise de áreas específicas de interesse;
- a estrutura de custos deve ser compatível com os procedimentos gerenciais e contábeis utilizados na coleta dos dados de custos.

Pilla (2003) propôs uma estrutura geral para a identificação dos elementos de custo que irão compor a EDC. Cada elemento de custo é formado por uma associação entre um produto, uma atividade executada em relação ao produto, um recurso consumido na atividade, um órgão que gera a atividade e um órgão que executa a atividade. Essa estrutura permite avaliar o impacto das características de construção, utilização e manutenção de um sistema técnico, em termos de localização de instalações de operação e de apoio e de políticas de manutenção e abastecimento, no custo total de posse.

Brick e Borges (2004) acrescentaram a essa estrutura as fases da vida útil do sistema técnico: Concepção e Definição, Projeto e Desenvolvimento, Fabricação/Construção, Utilização, Atualização/Modernização e Desativação.

Uma atividade pode ser definida como um conjunto de ações a serem praticadas sobre determinado produto, envolvendo recursos e tempo para a sua execução. Entre as ações executadas, podem ser encontradas outras atividades, permitindo uma organização hierárquica das mesmas.

As atividades do nível mais analítico na estrutura hierárquica são compostas por tarefas, que são unidades mais elementares do trabalho a ser executado para o alcance de um determinado fim.

Brick e Borges (2004) fizeram um levantamento dos tipos de atividades mais comuns usadas em análises de CVU, tendo chegado à seguinte relação:

- Gerenciamento: ato de planejar, dirigir e controlar;
- Pesquisa, Estudos e Análises: investigações com o intuito de adquirir informação e exames detalhados na tentativa de descobrir e conhecer mais sobre um determinado assunto;
- Simulação: utilização de modelos para representação simplificada de eventos mais complexos do mundo real;
- Engenharia: utilização de conhecimentos científicos e empíricos para o desenvolvimento e aplicação de tecnologia. Foi identificada a necessidade dos seguintes complementos para uma melhor definição desta atividade:
 - Projeto; e
 - Desenvolvimento.
- Aquisição: aquisição de produtos/sistemas já existentes no mercado;
- Fabricação: fabricação/construção de produtos/sistemas;
- Integração: combinação harmoniosa de elementos diversos em um produto ou sistema;
- Teste: verificação do funcionamento de um determinado item, produto ou sistema, e do seu grau de eficácia;
- Treinamento: ato de treinar, capacitar e qualificar pessoas. Foi identificada a necessidade dos seguintes complementos para uma melhor definição desta atividade:
 - Treinamento para Operação
 - Treinamento para Manutenção
- Instalação: ação de posicionar um item, produto ou sistema em um determinado local e prepará-lo para a utilização;
- Comissionamento: certificação do funcionamento de um item, produto ou sistema logo após a sua instalação;
- Operação: funcionamento de um dado item, produto ou sistema;
- Manutenção: ação de manter ou conservar um item, produto ou sistema em condições de funcionamento. Admitem-se os seguintes complementos para sua melhor identificação:
 - Preventiva;
 - Corretiva;
 - Sob condição;
 - Paradas; e
 - Outras.
- Desativação: ato de tornar o sistema inativo;
- Descarte: alienação ou reciclagem do item, produto ou sistema; e
- Perda de produção: parada ou redução de produção causada por eventos que causem indisponibilidade.

A essas, pode-se acrescentar as atividades de:

- Transporte: ação de movimentar um recurso entre duas localizações geográficas distintas;
- Abastecimento: ação de fornecer um insumo (matéria prima, componente ou sobressalente) para uma atividade de operação ou manutenção; e
- Armazenagem: ação de guardar e preservar um produto para posterior utilização.
- Atualização ou Modernização: ação de introduzir modificações nas instâncias do sistema, após as mesmas terem sido colocadas em operação. Essas atividades normalmente são precedidas de outras de gerenciamento, estudos, simulações, engenharia e aquisição.

Um aspecto importante a considerar é que o produto (sistema técnico, ou componente de um sistema técnico), ao qual as atividades devem se associar para compor um elemento de custo, pode possuir duas naturezas distintas. Inicialmente, as atividades são executadas no nível de projeto e desenvolvimento, quando o sistema ainda é uma entidade virtual, sem unidades fabricadas. Após o desenvolvimento do sistema (construção de um protótipo), muitas unidades (instâncias, ou sistemas

reais) do mesmo podem ser produzidas e operadas em locais distintos. Essas instâncias passam, então, a ter uma vida própria, estando sujeitas a falhas decorrentes da operação, gerando ações de manutenção, indisponibilidade e perdas de produção.

As atividades que se aplicam ao sistema em sua fase virtual são as de Gerenciamento, Pesquisa, Estudos e Análises, Simulação, Engenharia e Teste.

As atividades relacionadas às instâncias do sistema podem ser divididas em três grandes grupos, segundo as fases mais características do ciclo de vida do sistema (PILLA, 2003):

- a) Na fase de Produção/Construção, os equipamentos para manufatura, missão principal e apoio, a infra-estrutura e os softwares previstos no projeto são adquiridos, integrados, avaliados e testados. São realizadas, também, as atividades que pertinem à preparação do sistema para sua introdução em serviço (transporte, instalação, comissionamento, treinamento, etc.).
- b) Na fase de Operação e Apoio são executadas as atividades finalísticas de operação e as atividades de apoio logístico (que têm como objetivo manter as condições de disponibilidade do sistema principal, provendo todos os recursos necessários ao seu funcionamento e manutenção).
- c) A fase de Retirada de Operação e Descarte representa o final do ciclo de vida e ocorre quando as características do projeto já não permitem o atendimento dos objetivos ou quando a sua continuidade se torna economicamente inviável.

Atividades de Atualização e Modernização também podem ser realizadas em instâncias do sistema, podendo ser precedidas das mesmas atividades que se aplicam ao sistema virtual durante a sua fase de desenvolvimento.

3. EVENTOS GERADORES DE CUSTO

Segundo Pilla (2003) entende-se por “evento” uma ocorrência de execução de uma “atividade”, relacionada a um “produto”, num determinado instante do tempo do ciclo de vida. Um “modelo de geração de eventos” é utilizado na previsão dos instantes de tempo e dos locais de ocorrência dos diversos eventos relacionados a uma atividade ou, de uma forma mais agregada, da quantidade de eventos a serem executados num determinado intervalo de tempo.

A definição do modelo de geração de eventos é importante para a distribuição das atividades ao longo do espaço e do tempo e, conseqüentemente, dos custos relacionados às mesmas. Essas ações ocorrem, na maior parte das vezes, de forma programada, mas também podem ocorrer em instantes aleatórios. Algumas dessas atividades, além de ocorrerem em instantes aleatórios, também dependem do uso efetivo que se faz do sistema, como é o caso das manutenções corretivas geradas por falhas. Este fato aumenta a complexidade dos modelos de geração de eventos, pois eles irão depender do histórico da vida de cada instância do sistema.

Uma instância de sistema, ao longo de sua vida, poderá se encontrar em vários estados possíveis.

O sistema pode estar indisponível, quando não estiver em condições de produzir, o que pode ter sido causado por uma parada prevista, ou ocasionado por uma falha que impeça o funcionamento do sistema.

O sistema estará disponível, quando estiver em condições de produzir. Neste estado, o sistema poderá estar parado (em espera), ou operando. Logo, uma instância qualquer do sistema apresenta variações em seu estado ao longo de sua existência, podendo ocupar três estados básicos:

- Estado 0: Indisponível (capacidade produtiva nula);
- Estado 1: Disponível em Espera (capacidade produtiva de 100%, mas em “Stand by”); e
- Estado 2: Disponível Operando (funcionando com capacidade produtiva de 100%)

Outros estados intermediários, representando níveis diversos de capacidade produtiva (em vez de apenas duas alternativas: 0% ou 100%), podem ser estabelecidos, mas isto complicaria em muito a solução do problema, pois a quantidade de estados possíveis cresceria exponencialmente com a quantidade de níveis de produção que for considerada.

A maior parte dos eventos que geram atividades consumidoras de recursos e que afetam o CVU, independe do uso efetivo que se faz das unidades do sistema possuídas pela organização e dos estados em que elas se encontram. Como exemplos desses eventos pode-se citar aqueles relacionados às atividades de projeto, desenvolvimento, aquisição, fabricação, instalação, comissionamento, atualização e desativação. Os instantes de ocorrência desses eventos podem, portanto, serem medidos em uma escala de tempo corrido (Tempo Calendário).

Entretanto, para as atividades que dependem do uso do sistema, tais como muitas atividades de manutenção, os instantes de ocorrência dos eventos iniciadores dessas atividades devem ser medidos em escalas de tempo que caracterizem os períodos de uso efetivo do sistema (Tempo de Uso Efetivo).

Assim, convém definir dois tipos de escalas de tempo a serem utilizados:

- a) Escalas de Tempo Calendário; e
- b) Escalas de Tempo de Uso Efetivo.

As escalas relacionadas ao uso dependerão da forma como o sistema será utilizado, podendo haver a contabilização por horas de operação, quilômetros percorridos, e/ou por tarefas pontuais executadas, como pousos e decolagens. Neste último caso, a variável de uso a ser contabilizada é a simples execução da tarefa em um instante específico.

Os instantes dos eventos medidos em uma escala de tempo de uso efetivo devem ser convertidos para a escala natural (de calendário) com a inclusão dos tempos de indisponibilidade e espera. Assim, no que concerne aos eventos relacionados ao uso, torna-se importante o acompanhamento dos estados do sistema ao longo de todo o ciclo de vida, de forma a determinar os instantes em que o mesmo está disponível e operando e, conseqüentemente, acumulando desgaste dos componentes, o que levará à ocorrência de falhas e necessidade de manutenções. Especial atenção deve ser dada, então, aos eventos que ocasionam mudanças de estado. Estes eventos são os de aquisição e desativação dos produtos, de ocorrência de falhas ou paradas que causem indisponibilidade, de término de ações de manutenção executadas durante períodos de indisponibilidade e que restaurem o produto ao estado de disponível e os eventos de início e término de operação.

É importante destacar que, em uma análise de custo de vida útil, poderão existir muitas instâncias do sistema sendo usadas, cada uma delas com um perfil de operação próprio e, conseqüentemente, com escala de tempo de uso efetivo, específica. Assim, cada uma das instâncias do

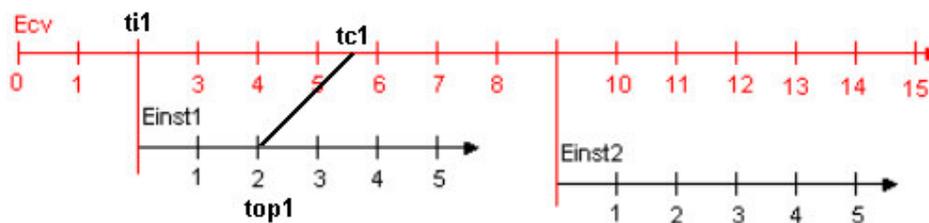


Figura 1: Escalas de Tempo para o Ciclo de Vida e duas Instâncias.

sistema em operação apresentará uma história de vida diferente das demais.

A figura 1 ilustra a conversão, entre as duas escalas, da medida do instante de ocorrência de um evento na instância 1 de um sistema.

Na figura foi usada a seguinte notação:

Ecv = escala de tempo calendário.

Einsti = escala de tempo de uso da instância i.

t_{ii} = instante de início de operação da instância i do sistema, medido na escala de tempo calendário.

top_i = instante de ocorrência de um evento medido na escala de tempo de uso da instância i.

Para um dado instante top_i , medido na escala $Einst_i$, obtém-se o tempo t_{ci} , na escala de tempo de calendário (Ecv), com a adição de t_{ii} e do somatório dos tempos em que a instância permaneceu nos estados 0 (indisponível) e 1 (em espera), representado na figura como a projeção da reta top_1-tc_1 sobre o eixo do tempo calendário.

Para as atividades que independem do uso, o modelo de geração de eventos consiste em uma simples programação pré-determinada (Planos), dos instantes de ocorrência desses eventos, medidos em tempo calendário. Essa programação poderá ser feita para cada tipo de atividade, independentemente das demais, mesmo que os instantes de ocorrência desses eventos e as durações das atividades por eles geradas sejam aleatórios.

As principais programações que se enquadram nessa categoria são as seguintes:

- Plano de Atividades de Engenharia, relacionado às atividades de gerenciamento, estudos, simulação, concepção, projeto e desenvolvimento do sistema, ainda em sua fase virtual;

- Plano de Aquisição, que define os instantes em que serão adquiridas/fabricadas as instâncias do sistema e adquiridos os estoques de sobressalentes;
- Plano de Instalação, que define as datas e os locais de colocação em operação das instâncias do sistema.
- Plano de Operação, que determinará o regime de operação das instâncias do sistema, durante seu ciclo de vida (supondo que a operação não será reprogramada em função de indisponibilidades do sistema);
- Plano de Manutenção por Calendário, referente à programação das manutenções preventivas por tempo calendário;
- Plano de Atualização, referente ao projeto e introdução de modificações a serem efetuadas na estrutura e funcionamento das instâncias do sistema ao longo de seu ciclo de vida.
- Plano de Desativação, com a programação das datas previstas para a desativação das instâncias do sistema.

No entanto, para aquelas atividades ligadas a eventos que causam indisponibilidade e dependem do uso, tais como manutenções corretivas, perda de produção, tratamento de situações de acidentes e manutenções preventivas por uso, não é possível elaborar, previamente, um plano de geração de eventos nos mesmos moldes dos planos acima descritos. Isto porque essas atividades ao mesmo tempo em que dependem da linha de vida, também são elementos que afetam a linha de vida. Uma grande parte dessas atividades é gerada por eventos de falha nos componentes do sistema. A análise de falhas representa, portanto, um capítulo à parte e muito importante da ACVU.

Após a instalação e montagem, as instâncias do sistema passam a operar no sentido de executar a missão determinada para as mesmas, sofrendo desgaste pela própria utilização e levando à ocorrência de diversas falhas, as quais deverão ser corrigidas por ações de manutenção. Este ciclo de operações e manutenções ocorre durante toda a fase de utilização do sistema, até o momento do descarte da instância.

Segundo Pilla (2003), a ocorrência de uma falha pode afetar de diversas formas o desempenho, não só do “produto” onde ocorre, mas, também, do sistema como um todo. Exemplos desses impactos são:

- Perda de capacidade de produção;
- Indisponibilidade do sistema ou dispositivo;
- Geração de outras falhas;
- Danos materiais aos componentes da estrutura de operação e apoio;
- Danos a seres humanos e ao meio-ambiente.

Define-se como “modos de falha” as diferentes formas com que um componente pode falhar ao executar determinada função do sistema. Como um sistema é composto de diversos componentes, e os mesmos podem apresentar diferentes “modos de falha”, estas falhas podem influir no funcionamento, não apenas do componente, mas do sistema como um todo. Pilla (2003) apresenta um modelo para o “modo de falha”, usando a notação da UML (Unified Modeling Language), onde associa o mesmo a um produto e relaciona, em um segundo nível, as falhas de um produto a atividades que as causam, ou que deverão ser executadas para corrigi-las. Uma versão modificada deste modelo pode ser observada na figura 2. Uma outra característica do modelo original de Pilla, não mostrada na figura, é a de que o mesmo inclui a possibilidade de se considerar a ocorrência de atividades que previnem modos de falha (tais como manutenções preventivas). No modelo simplificado aqui representado, algumas atividades sobre determinados produtos podem gerar modos de falha e estes modos de falha geram novas atividades. Um exemplo dessa situação é a operação do sistema, que causa falhas no funcionamento dos componentes, gerando demanda para órgãos mantenedores, os quais deverão efetuar atividades no sentido de restabelecer a operação do sistema inicialmente citado.

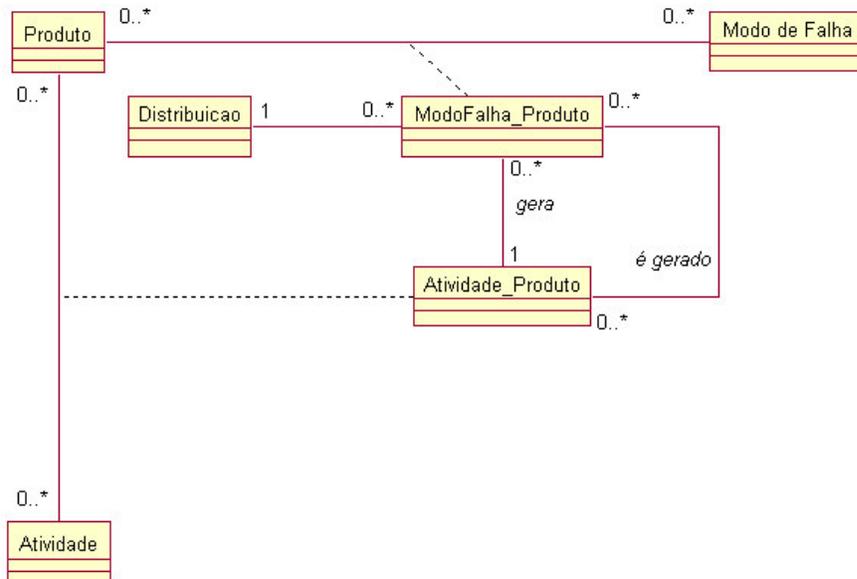


Figura 2: Diagrama de Classes Alterado para Modos de Falha.

5. LINHAS DE VIDA DE INSTÂNCIAS DE SISTEMAS TÉCNICOS

O estado das instâncias do sistema depende dos eventos geradores de indisponibilidade e dos eventos de início e fim de operação. Portanto, o regime de operação dos sistemas exerce um papel importante na ACVU.

Sistemas distintos podem apresentar regimes diferentes de operação. Os tipos básicos de regimes de operação são os seguintes:

- a) por missão completa;
- b) operação contínua; e
- c) operação agendada.

No regime de missão completa há um roteiro determinado de tarefas a serem cumpridas pelo sistema, caracterizando ciclos completos, tais como viagens de avião e navio. Isto os diferencia de sistemas com operação contínua (também denominada operação 24/24), onde existe apenas uma missão, sendo cumprida ao longo de todo o ciclo operacional do sistema, como em sistemas para extração de petróleo.

A operação agendada caracteriza-se por ciclos de operação com início e fim definidos, repetindo-se segundo regras pré-definidas. Exemplo deste tipo de operação ocorre em instalações industriais de transformação, onde há um horário pré-definido para o seu funcionamento.

Nos casos de operação por missão completa ou agendada, deve-se também considerar a possibilidade de deslocamento ou cancelamento da operação do sistema, no caso de haver uma indisponibilidade qualquer. A Figura 2 ilustra o histórico dos estados de um sistema operando em um desses dois regimes, sem interrupções no seu funcionamento.

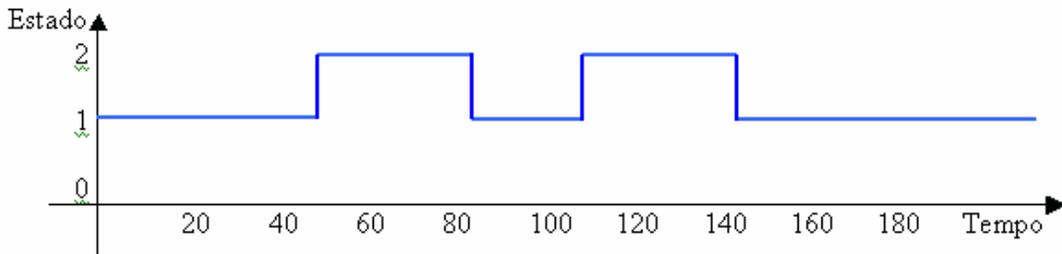


Figura 3: Operação Normal, sem eventos que causem indisponibilidade.

Os eventos podem ser modelados como pulsos ocorrendo em determinados instantes da existência do sistema, com as amplitudes dadas pela duração das atividades a eles associadas. Tanto os instantes de ocorrência, como as durações das atividades, podem ser determinísticos ou aleatórios.

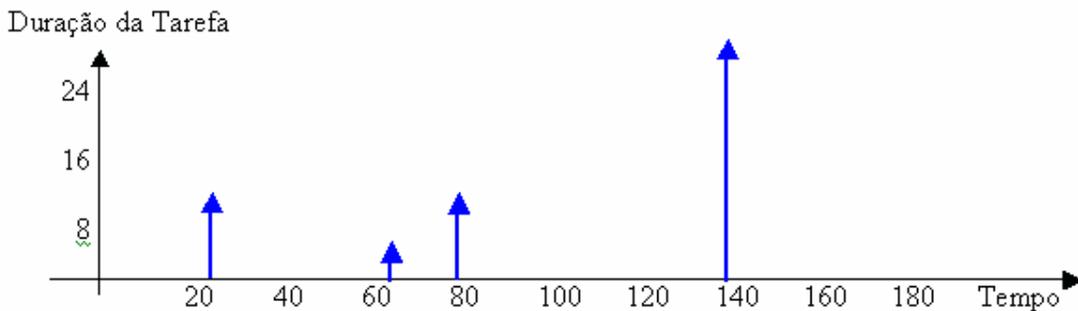


Figura 4: Eventos com tempos de ocorrência e de duração aleatórios.

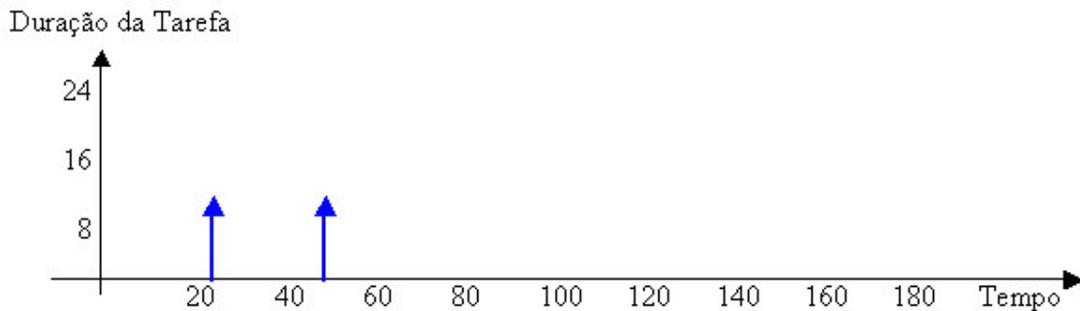


Figura 5: Eventos com tempos de ocorrência e duração determinísticos.

O impacto da indisponibilidade na operação do sistema pode ser considerado de duas maneiras:

- a. Deslocamento em uma faixa do gráfico de operação do sistema.
- b. Cancelamento de trechos de operação.

Uma ilustração da primeira hipótese, considerando a operação normal mostrada na figura 3 e os eventos causadores de indisponibilidade da figura 5, é apresentada na figura 6, enquanto a figura 7 ilustra a segunda hipótese, sob as mesmas condições.

A linha de vida não é aplicável apenas a instâncias do sistema, mas também a qualquer produto componente do mesmo. Uma informação a ser adicionada à linha de vida, tanto para componentes quanto para sistemas móveis, é a localização do mesmo. Dentro de uma estrutura de manutenção, o acompanhamento ponto-a-ponto da localização do componente é extremamente

importante para o cálculo do custo de transporte e para a escolha do estoque a ser utilizado para a reposição do componente.

Um componente pode apresentar basicamente os mesmos estados definidos para o sistema: disponível para operar, operando e indisponível. Todo componente apresenta-se pronto para operar a

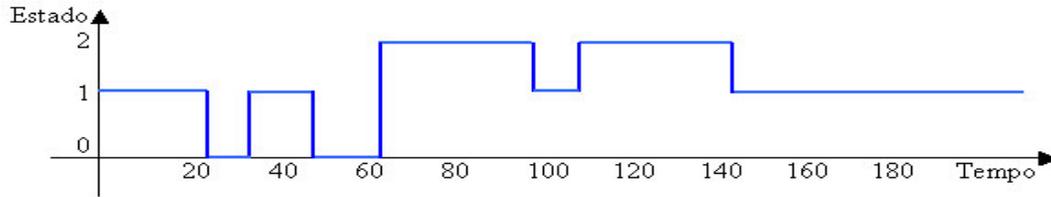


Figura 6: Efeito de indisponibilidade, com deslocamento de operação.

partir de sua aquisição, mas apenas irá operar junto ao sistema, após ser instalado no mesmo. Quando o componente é utilizado no sistema, ele passa a contabilizar horas de operação (ou outra variável que

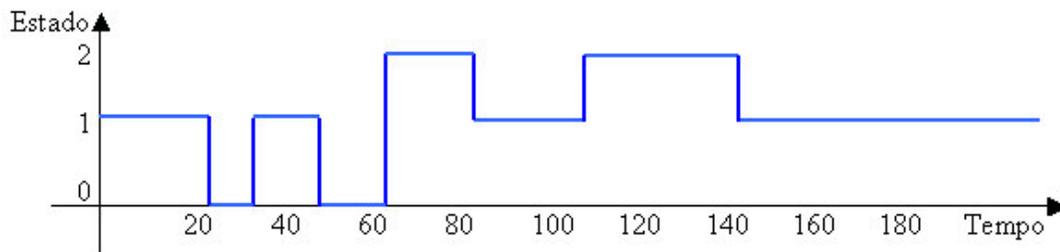


Figura 7: Efeito de indisponibilidade, sem deslocamento de operação.

indique o seu uso, como quilômetros rodados, milhas navegadas, ou pousos e decolagens efetuadas) que levam às necessidades de manutenção corretiva e preventiva devido ao uso, segundo os modos de falha do componente.

Quando da ocorrência do modo de falha, o componente passa ao estado de indisponível, solicitando uma ação de manutenção no sentido de levá-lo de volta ao estado de disponível para operar.

Tendo em vista a dependência circular existente entre eventos de início e término de operação e de falhas, torna-se necessária definir um processo para geração dos eventos, de forma a organizar a distribuição dos eventos citados. Uma possibilidade é a definida pelos seguintes passos gerais:

- a) Gerar Planos de Aquisição e Instalação.
- b) Gerar Plano de Operação supondo 100% de disponibilidade
- c) Ajustar Plano de Operação em função de grandes indisponibilidades programadas em tempo calendário (paradas e atualizações).
- d) Ajustar plano de operação para as indisponibilidades que dependem da própria operação (manutenção preventiva/corretiva, acidentes).
- e) Repetir o processo a partir do passo c) até atingir um critério de “coerência”.

Ao final desta etapa, são obtidos eventos para as seguintes atividades relacionadas a cada uma das instâncias do sistema:

- a) aquisição
- b) manufatura
- c) integração
- d) instalação
- e) montagem
- f) comissionamento
- g) operação

- h) manutenções corretivas / preventivas
- i) treinamentos
- j) paradas / atualizações
- k) acidentes
- l) desativação

6. ALGORÍTMOS PARA IMPLEMENTAÇÃO DE LINHA DE VIDA E GERAÇÃO DE EVENTOS

Segundo Drozdek (2002), uma estrutura ligada (ou lista encadeada) é uma coleção de nós, que armazenam dados, e de ligações com outros nós. Isto possibilita o armazenamento de dados em qualquer lugar da memória, e a passagem de um nó para outro da estrutura ligada é efetuado com a utilização dos endereços armazenados nos próprios nós.

Em termos computacionais, a linha de vida pode ser representada por uma estrutura de armazenamento de dados do tipo lista cujos nós (N) armazenam o estado (Es) e o instante de transição (Td) para o mesmo, conforme a figura 8. Esses nós serão referenciados como nós de estado de uma instância do sistema.

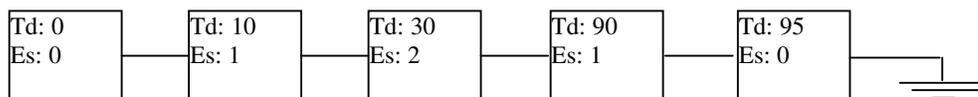


Figura 8: Exemplo de Lista de Linha de Vida

Esta lista considera que o estado inicial e final de um sistema será sempre indisponível, pois se referem, respectivamente, às atividades anteriores ao primeiro ciclo operacional e às atividades referentes ao final do ciclo de vida, como descarte. Como esta é uma lista simplesmente encadeada, utilizar-se-á como terminador um elemento nulo para indicação do final da mesma.

Seguindo a notação de lista, em trechos subseqüentes deste texto, para referenciar um determinado campo de um dado nó N, será utilizada a notação N.campo, o que permite expressar o estado do nó como N.Es e o instante de mudança como N.Td. Para referenciar o nó seguinte da lista, utiliza-se N.posterior, o que significa que o instante de mudança para o próximo estado seria expresso por N.posterior.Td.

Algoritmos para determinar o nó de uma lista cujo tempo de transição para o seu estado seja imediatamente anterior ou posterior a um tempo qualquer Tc, denominados, respectivamente, Anterior(Tc) e Posterior(Tc), podem ser facilmente construídos.

O algoritmo Posterior(Tc) pode ser descrito matematicamente como $Td = \text{Min}(Td) \mid Td \geq Tc$.

1. Selecionar o primeiro nó da lista em N1
2. Definir N0 como um elemento nulo
3. Se N1 é um elemento nulo, ou $N1.Td \geq Tc$, definir $N0 = N1$,
Senão,
N1 recebe o próximo nó da lista e repete o passo 3
4. Retornar N0

Quanto ao Anterior(Tc), pode ser descrito como $Td = \text{Max}(Td) \mid Td < Tc$.

1. Selecionar o primeiro nó da lista em N1
2. Definir N0 como um elemento nulo
3. Se N1 é um elemento nulo, ou $N1.Td > Tc$, ir diretamente para o passo 6
4. Definir N2 como o próximo nó da lista a partir de N1 (N1.próximo)

5. Se $N2$ é um elemento nulo, ou $N2.Td > Tc$, definir $N0 = N1$,
Senão,
N1 recebe N2 e repete o passo 3
6. Retornar N0

O primeiro passo do processo delineado na seção anterior implica na geração de uma lista incluindo todos os eventos relacionados a atividades de aquisição e instalação de instâncias e sua operação pré-programada, sem considerar interrupções para manutenções ou atualizações.

A seguir, devem ser consideradas as grandes paradas, atualizações e manutenções previstas em calendário e que causem indisponibilidades e calculado o seu efeito sobre a operação do sistema, gerando a linha de vida programada do mesmo. A regra estabelecida é a de que sempre haverá o cancelamento de operações nestes casos.

Assim como a linha de vida, as indisponibilidades programadas também podem ser armazenadas em listas, contendo o instante e a duração de cada evento, como na figura 9. Nesta estrutura de lista, os nós (aqui denominados nós de parada) expressam o instante de início (Tc) e a duração da parada (Dc).

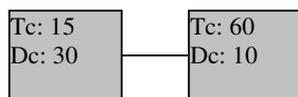


Figura 9: Lista de Paradas Previstas

Ao determinar o efeito de uma lista de paradas previstas sobre uma lista de linha de vida representando a operação normal, deverá ser obedecida a seguinte seqüência, para cada nó de parada (Tc e Dc):

1. Criar um nó de estado $N0$ com $Td = Tc$, e $Es = 0$
2. Definir $Tf = Tc + Dc$
3. Localizar o nó de estado $N1$ da lista de linha de vida com Td imediatamente anterior a Tc , usando o algoritmo Anterior(Tc)
4. Localizar o nó de estado $N2$ da lista de linha de vida com Td imediatamente posterior a Tf , usando o algoritmo Posterior(Tf)
5. Definir $t1 = N1.Td$, e $t2 = N2.Td$
6. Se $\text{Anterior}(t2) = N1$, não há nós entre $N1$ e $N2$, o que significa que o estado entre os instantes $t1$ e $t2$ é constante.
 - a. Se o estado de $N1$ é indisponível, termina o processo
Senão,
Definir $N3$ como um novo nó de estado
7. Se $\text{Anterior}(t2)$ diferente de $N1$, significa que existem nós entre $N1$ e $N2$, e $N3$ deverá ser iniciado com o nó Anterior($t2$).
8. Para $N3$, definir $Td = Tf$, $Es = \text{Anterior}(t2).Es$, e próximo = $N2$
9. Para $N1$, definir próximo como $N0$
10. Para $N0$, definir próximo como $N3$

Ao final desse processo, a lista de linha de vida englobará as paradas programadas em tempo calendário.

Para os eventos que dependem do uso e causam indisponibilidade, o ajuste deve ser feito de forma iterativa. Isto é, determina-se o período de indisponibilidade causado pelo evento, ajusta-se a linha de vida da instância do sistema e calcula-se o novo período de indisponibilidade a partir da nova linha de vida ajustada e assim recursivamente até o final da vida da instância do sistema.

A lista de linha de vida citada anteriormente utiliza o instante de tempo de calendário para instância de sistema, associado ao estado do sistema-produto a partir daquele instante. Uma informação importante, obtida a partir desta lista, é a duração de um determinado estado, pois para um nó N , a duração daquele estado é dada por:

$$\text{Duração}(N) = N.\text{próximo}.T_d - N.T_d$$

Com isto, pode-se calcular o tempo total de operação do sistema através do somatório das durações dos nós onde o estado é operando ($E_s=2$). Esta informação será importante para o cálculo dos instantes onde ocorrerão as manutenções por uso.

Para efetuar uma conversão do tempo de uso para o tempo de calendário, a partir da linha de vida, deve-se percorrer a mesma considerando-se a duração acumulada de todos os nós onde o estado tem valor 2. O algoritmo definido para esta conversão é explicitado a seguir:

1. Definir T_o como o tempo operacional fornecido
2. Definir t_c como o tempo de calendário desejado
3. Iniciar t_c com o valor -1 (t_c não encontrado na lista)
4. Iniciar $N1$ como o primeiro nó da lista
5. Definir o acumulador Ac , iniciado com valor 0. Ac irá receber o tempo operacional acumulado ao final de cada nó com estado 2.
6. Enquanto $N1$ for diferente do terminador nulo, faça:
 - a. Se o estado de $N1$ tem valor 2, incrementar Ac com $\text{Duração}(N1)$
 - b. Se Ac é superior ou igual a T_o , $t_c = N1.T_d + \text{Duração}(N1) - (Ac - T_o)$
 - c. $N1$ recebe o próximo nó ($N1=N1.\text{próximo}$)
7. Retornar o valor t_c

Como um sistema é composto de vários componentes que apresentam modos de falha com diferentes distribuições, para se determinar o instante em que a instância do sistema ficará indisponível, deve-se determinar qual será o primeiro componente a apresentar uma falha decorrente do uso e causadora de indisponibilidade, a partir de determinado instante.

Esta falha deverá ser corrigida, de forma que o sistema volte a operar, mas causará um ajuste, com o cancelamento de parte da operação do sistema, em sua linha de vida. Este ajuste terá uma influência direta sobre a determinação do instante para a próxima falha, haja vista que deslocará o período onde será completado o tempo (ou quantidade) de operação estimado para a falha.

Como pode ser observado, trata-se de um processo recursivo, onde a instância do sistema sofre o efeito de uma falha, corrige sua linha de vida, e repete o processo para a próxima falha prevista, até o momento em que há o final do ciclo de vida do sistema.

Seja $T_{Fi}(\text{top})$ o intervalo de tempo previsto, a partir do tempo de operação top , para a ocorrência de uma falha em componente de uma instância do sistema, causadora de indisponibilidade no sistema. O instante da primeira falha da instância do sistema, na escala de tempo de uso efetivo (tf_0), pode ser determinado pelo menor tempo de falha entre todos os previstos a partir do instante $\text{top}=0$:

$$tf_0 = \text{Min}(T_{Fi}(0))$$

Após definir a indisponibilidade para o instante tf_0 , com duração equivalente à duração total das tarefas relacionadas à manutenção do modo de falha, define-se um evento gerador de indisponibilidade na linha de vida do sistema, onde D_c equivale à duração supracitada e o instante de transição é obtido com a conversão do tempo operacional para tempo de calendário, ajustando-se a linha de vida do sistema a partir deste evento, e repetindo-se o processo a partir do instante tf_0 para o cálculo de tf_1 , e assim em diante, de forma recursiva, até o momento em que o tempo calculado supera o tempo total de operação do sistema:

$$tf_n = \text{Min}(T_{Fi}(tf_{n-1}))$$

Quando houver coincidência dos instantes de falhas entre vários modos de falha, este modelo irá considerar o máximo dos tempos de todas as tarefas necessárias para a manutenção de todos os modos de falha que ocorreram, de forma a calcular o valor de D_c naquele ponto.

Ao final de todas as iterações, o resultado será uma lista de linha de vida que expressa a toda a história de vida do sistema, com todas as simulações referentes à operação e manutenção do mesmo.

Os eventos de manutenção que causam indisponibilidade, a partir do tempo de operação, podem ser armazenados em uma lista contendo o instante operacional (T_o) e a duração de cada evento (D_c), como na figura 9.

Da mesma forma que era possível calcular o nó posterior a um dado instante de calendário,



dentro de uma lista de linha de vida, será possível determinar o nó posterior a um dado tempo de

Figura 11: Lista de Eventos por Uso

operação (top), dentro de uma lista de eventos por uso, conforme o algoritmo:

1. Selecionar o primeiro nó da lista em N1
2. Definir N0 como um elemento nulo
3. Se N1 é um elemento nulo, ou $N1.To > top$, definir $N0 = N1$,
Senão, N1 recebe o próximo nó da lista e repete o passo 3
4. Retornar N0

Para efetuar a correção da linha de vida, para cada instante de operação em que ocorra um ou mais eventos de falha causadora de indisponibilidade, deverá ser observado o tempo máximo gasto com a correção destas falhas. Obtendo-se o tempo máximo, este será aplicado à linha de vida do sistema-produto no instante de calendário equivalente. Isto é expresso no algoritmo abaixo:

1. Definir um vetor V de listas de eventos por uso
2. Definir T_i como o tempo de operação inicial, iniciando com valor 0
3. Calcular a partir do tempo de operação inicial o primeiro elemento de cada lista contida no vetor V
4. Definir N1 como Posterior(T_i) para o primeiro elemento de V
5. Para os demais elementos de V faça:
 - a. Definir N2 como Posterior(T_i) para o elemento corrente de V
 - b. Se $N2.To$ for menor que $N1.To$, assumo $N1 = N2$
6. Definir o Acumulador Ac com valor inicial 0
7. Para cada elemento de V faça:
 - a. Definir N2 como Posterior(T_i) para o elemento corrente de V
 - b. Se $N2.To$ equivale a $N1.To$:
 - i. O próximo nó de evento para o elemento corrente é calculado e acrescido ao final da lista
 - ii. Se $N2.Dc > Ac$, Ac recebe o valor de $N2.Dc$
8. Converter $N1.To$ para tempo de calendário (armazenar em T_c)
9. Se T_c obtém valor -1, foi extrapolada a linha de vida do sistema e o processo de correção deve ser encerrado neste ponto
10. Determinar um evento calendário (pseudo-evento) com instante T_c e duração Ac
11. Aplicar o evento calendário sobre a linha de vida do sistema
12. Mudar o valor de T_i para $N1.To$
13. Repetir o processo a partir do item 4

Ao final deste processo, a linha de vida programada do sistema será corrigida para as indisponibilidades geradas por eventos medidos em escalas de uso, determinando a linha de vida efetiva do mesmo, e a lista irá armazenar toda a seqüência de eventos medidos em escalas de uso e calendário, que ocorrerão ao longo da vida do sistema.

7. CONCLUSÕES

A Análise de Custo de Vida Útil (ACVU) implica na identificação de todas as atividades consumidoras de recursos constituintes da Estrutura de Desdobramento de Custo (EDC) e dos seus respectivos instantes de ocorrência.

Um sistema de cálculo de ACVU deve, pois, ser capaz de definir estes instantes através de modelos de geração de eventos, considerando tanto as atividades agendadas em tempo calendário, quanto as atividades que dependem do uso efetivo das instâncias do sistema, levando em consideração que a ocorrência de um determinado evento causador de indisponibilidade irá afetar os instantes de ocorrência de outros eventos.

Como visto, isto é feito pelo acompanhamento da utilização do componente através de sua linha de vida, e de rotinas geradoras de números aleatórios, visando satisfazer às distribuições de falhas, de acordo com as características dos componentes utilizados. Mais do que isto, observa-se a necessidade do relacionamento entre os modos de falha do produto e as atividades necessárias para o restabelecimento de suas funções.

Com o completo mapeamento das atividades executadas ao longo da vida do sistema, o cálculo do custo de vida útil será dado pelo valor total do somatório dos custos descontados, com o uso de uma taxa de desconto apropriada, dos recursos necessários à execução das mesmas. O agrupamento desses custos segundo a EDC permite a visualização de gastos segundo vários critérios: por componentes do sistema, por órgãos geradores ou executores de atividades e por fases do ciclo de vida.

Para uma implementação mais eficaz deste processo de cálculo, levando em consideração a grande quantidade de possibilidades de concepção para os mais diferentes sistemas e a grande diversidade de características para os seus componentes, deve ser utilizada uma ferramenta com grande poder de expansibilidade, de forma a se adequar de forma gradativa aos diferentes elementos considerados a cada análise.

O modelo descrito neste artigo foi utilizado para implementar uma ferramenta computacional para ACVU, com as características apontadas acima. Esta ferramenta será descrita em um outro artigo, a ser publicado posteriormente.

9. REFERÊNCIAS BIBLIOGRÁFICAS

Fuller, S. K. & Petersen, S. R. (1996) NIST Handbook 135 Life-Cycle Costing Manual for the Federal Energy Management Program. USA: National Institute of Standard and Technologies., 210p..

Pilla, L. H. L. (2003) - Um Modelo Conceitual para o Processo de Análise do Custo de CVU. Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal Fluminense, Niterói. 166p.

Brick, E.S. & Borges, R.C. (2004) Estrutura de Desdobramento de Custos para Análise de Custo de Vida Útil de Sistemas Técnicos. XXV ENEGEP, Florianópolis, SC.

Blanchard, B. S & Fabrycky W. J. (1998) System Engineering and Analysis. 3 ed. Prentice-Hall, United States of America.

Drozdek, A. Estrutura de Dados e Algoritmos em C++. 1 ed. São Paulo: Pioneira Thomson Learning, 2002. 580 p. Tradução de: Data structures and algorithms in C++.