

EARNED VALUE MANAGEMENT EM PROJETOS ÁGEIS DE SOFTWARE: ABORDAGENS DE APLICAÇÃO

Heitor Luiz Murat de Meirelles Quintella

UFF - Universidade Federal Fluminense

E-mail: hquintel@uninet.com.br

Isac Mendes Lacerda

UFF - Universidade Federal Fluminense

E-mail: isac.mendes@gmail.com

Resumo

Este artigo revisa brevemente os conceitos de *Earned Value Management* (EVM) e destaca quatro abordagens para seu uso em projetos ágeis de *software*. Desta maneira, este artigo se caracteriza como uma pesquisa exploratória de caráter analítico, temporal, de atualização e bibliográfico. Após a apresentação das quatro abordagens, uma comparação entre elas é feita, indicando características centrais, pontos fortes e limitações percebidas. O trabalho registra a preocupação em adaptar conceitos de EVM às características dos projetos ágeis de *software*, de modo que usufruam dos benefícios da técnica. Ao mesmo tempo, constata-se a necessidade de experiências adicionais para o amadurecimento e expansão da aplicação de EVM em projetos ágeis.

Palavras-chave: Earned Value Management (EVM), projetos ágeis de software, controle.

Abstract

This article briefly reviews the concepts of Earned Value Management (EVM) and highlights four approaches for its use in agile software projects. Thus, this article is characterized as an exploratory research and maintains analytical character, temporal, of update, and bibliographic. After the presentation of the four approaches, a comparison is made between them, indicating the central features, strengths and perceived limitations. The paper reports a concern in adapting the concepts of EVM to the characteristics of agile software projects, so enjoy the benefits of the technique. At the same time, there is the need for additional experiences for the maturing and expanding the application of EVM in agile projects.

Key Words: Earned Value Management (EVM), agile software projects, control.

1. Introdução

A natureza intangível e mutável do *software* é apontada como um fator que torna o gerenciamento do projeto de *software* fundamentalmente diferente do gerenciamento de outros tipos de projetos, como os da construção civil, por exemplo (MAQSOOD; JAVED, 2007; PRESSMAN, 2006). Em projetos de *software* os requisitos não estão necessariamente claros em fases iniciais (COCKBURN, 2002; LI *et al.*, 2008) e a medição de progresso de algo intangível traz consigo um conjunto próprio de dificuldades. Essa última característica possibilita a realização de avaliação subjetiva de avanço, por se tratar de algo intangível, que é apontada como uma das principais razões de falha na construção de *software* (JONES, 2006; BOOCH, 2003). Em outras palavras, embora não seja determinante, a intangibilidade pode contribuir para avaliações imprecisas de progresso de projetos.

Abordagens modernas de produção de *software* (denominadas ágeis) apontam a flexibilidade para incorporar mudanças nos projetos como ponto crítico para o sucesso (AGILE ALLIANCE, 2011; SCHWABER, 1996). Tendo esse argumento como base, entre outros, recentes autores têm sugerido adaptações na aplicação de *Earned Value Management* (EVM), uma técnica de monitoramento e controle genérica de projetos, para o uso mais alinhado às características dos projetos de *software* (RUSK, 2009; HANNA, 2009; CABRI; GRIFFITHS, 2006; SULAIMAN *et al.*, 2006).

Nesse contexto, o presente artigo registra quatro abordagens de aplicação, apresentadas nos últimos seis anos, que sugerem alguma adaptação dos conceitos de EVM para uma utilização mais aderente ao universo de *software*. Antes da apresentação dessas abordagens, o artigo destaca a metodologia utilizada e faz uma breve revisão dos conceitos de EVM. Na sequência, cada uma das abordagens é apresentada e é feita uma comparação entre elas, indicando características centrais, pontos fortes e limitações. Por último, são apresentadas as considerações finais do trabalho.

2. Metodologia

O propósito estabelecido para este trabalho foi identificar e comparar recentes abordagens na literatura sobre a aplicação de EVM em projetos ágeis de *software*. Desta maneira, este trabalho pode ser utilizado como referencial para pesquisas futuras no que se refere a monitoramento e controle de projetos de *software*.

Para cumprir tal propósito, este estudo tomou como ponto de partida a identificação de trabalhos nas bases credenciadas no Portal de Periódicos Capes. A busca de artigos assumiu como verdade que os trabalhos relevantes envolvendo o uso de EVM em projetos ágeis teriam uma das seguintes expressões no título: “*earned*”, “EVM” ou “EV”. Além disso, a expressão “*agile*” deveria estar mencionada em qualquer parte do texto dos artigos. Com esses critérios, foram encontrados apenas três estudos. A partir disso, um trabalho de busca complementar, utilizando os mesmos parâmetros, foi iniciado em revistas específicas sobre gerenciamento de projetos e onde normalmente são publicados trabalhos sobre EVM (ex.: *Project Management Journal*, *International Journal of Project Management*, *The Journal of Defense Software Engineering*, *International Journal of Managing Projects in Business* e *Leadership in Project Management*). O resultado desses esforços permitiu a identificação de um total de seis artigos. Desses seis, dois foram descartados por terem sido considerados pouco práticos e por trazerem ideias já contidas nos outros quatro trabalhos. Desta forma, foram feitas análises detalhadas e comparações entre as quatro abordagens selecionadas.

Essa estrutura permitiu ainda a classificação da pesquisa quanto ao propósito, abrangência, função e tipo de análise elaborada (NORONHA; FERREIRA, 2000). Assim, o artigo foi definido como:

- Quanto ao propósito (analítico ou de base) – analítico, por se tratar de uma pesquisa sobre aplicações de EVM em projetos ágeis de *software*.
- Quanto à abrangência (temporal ou temático) – temporal, por se tratar do relato de experiências de aplicação dentro dos últimos seis anos.
- Quanto à função (histórico ou de atualização) – de atualização, por se tratar de aplicações de EVM, considerando adaptações e práticas modernas de construção de *software*.
- Quanto ao tipo de análise (bibliográfico ou crítico) – bibliográfico, por se tratar de uma pesquisa na literatura.

3. *Earned Value Management* (EVM)

EVM é uma técnica universal de projetos criada a mais de 100 anos por engenheiros norte americanos, com a finalidade de comparar trabalho planejado, trabalho realizado e custo realizado. Entretanto, a primeira publicação do conceito só foi registrada na década de 60, a partir de uma experiência de sucesso em um projeto do governo dos Estados Unidos. Dos anos 60 aos anos 90 o setor privado manifestou pouco interesse e conhecimento

para o uso de EVM. Mas a publicação do padrão ANSI/EIA-748-1998 foi capaz de desmistificar vários aspectos do uso da técnica, o que despertou um grande interesse no setor privado (FLEMING; KOPPELMAN, 1998 e KIM *et al.*, 2003).

O uso de EVM permite avaliar o trabalho obtido em relação ao que foi gasto e ao que estava planejado gastar (VARGAS, 2003). Para Putz *et al.* (2007), EVM é uma técnica que permite ao gerente de projetos determinar as verdadeiras variações de prazo e custo de um projeto. Anbari (2003) afirma ainda que EVM não só revela variações de prazo e custo, como também permite fazer projeções e tomar decisões antes do fim do projeto.

EVM tem quatro elementos principais: *Earned Value* (EV), *Planned Value* (PV), *Actual Cost* (AC) e *Budget At Complete* (BAC). Suas descrições estão indicadas no quadro 1 abaixo:

Elemento	Descrição
<i>Earned Value</i> (EV)	É o valor orçado atribuído ao trabalho pronto de uma ou várias tarefas do projeto. EV também pode ser denominado como o valor monetário atribuído ao trabalho concluído.
<i>Planned Value</i> (PV)	É o custo orçado para o trabalho planejado de uma ou várias tarefas do projeto.
<i>Actual Cost</i> (AC)	É o custo real de uma ou várias tarefas do projeto.
<i>Budget At Complete</i> (BAC)	Valor orçado total do projeto.

Quadro 1 – Elementos Principais de EVM

Também é possível dizer que ao concluir uma tarefa o seu valor orçado passa a ser Valor Agregado ou *Earned Value* (EV) (LACERDA, 2009). A partir dos elementos principais do quadro 1 é possível derivar avaliações de prazo e custo, de acordo com os quadros 2 e 3.

Varição	Descrição
<i>Cost Variance</i> (CV)	Representa a diferença entre <i>Earned Value</i> (EV) e <i>Actual Cost</i> (AC) em uma data de referência do projeto [$CV = EV - AC$].
<i>Scheduled Variance</i> (SV)	Indica a diferença entre <i>Earned Value</i> (EV) e <i>Planned Value</i> (PV) em uma data de referência do projeto. SV se refere ao atraso ou antecipação do projeto em uma linguagem monetária [$SV = EV - PV$].
<i>Time Variance</i> (TV)	Indica o tempo entre o momento em que <i>Earned Value</i> (EV) foi medido e o momento em que deveria ter sido atingido quando comparado ao <i>Planned Value</i> (PV).

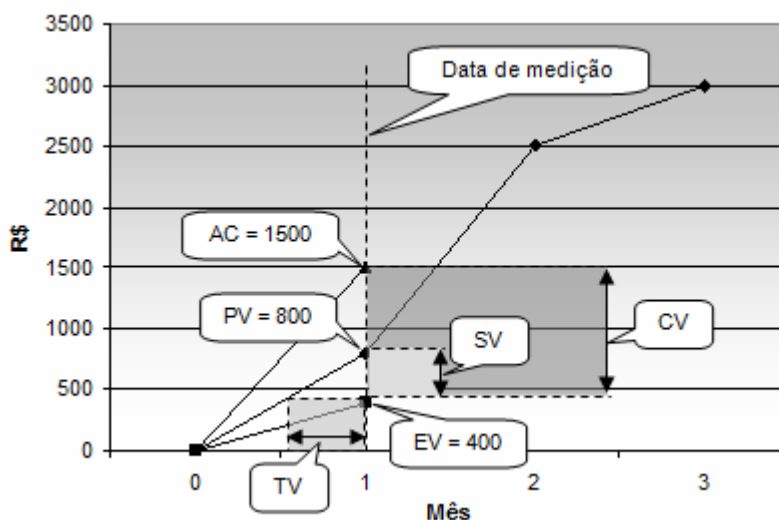
Quadro 2 – Variações de Prazo e Custo

Índice	Descrição
<i>Schedule Performance Index</i> (SPI)	Sinaliza percentualmente a realização do projeto comparado ao que estava previsto em uma data de referência. O SPI é determinado pela multiplicação de 100 pela divisão de <i>Earned Value</i> (EV) por <i>Planned Value</i> (PV) [$SPI = 100 * (EV / PV)$].
<i>Cost Performance Index</i> (CPI)	Sinaliza o percentual de eficiência dos custos do projeto em uma data de referência. O CPI é determinado pela multiplicação de 100 pela divisão de <i>Earned Value</i> (EV) por <i>Actual Cost</i> (AC) [$CPI = 100 * (EV / AC)$].

Quadro 3 – Índices de Prazo e Custo

Os elementos principais e as variações de prazo e custo de EVM podem ser avaliados na figura 1, que apresenta um projeto exemplo. De forma geral, pode-se dizer que ao final do primeiro mês, o projeto da figura 1 tem desvio de prazo e desvio de custo, pois EV é menor que PV e também menor que AC.

Figura 1 – Exemplo EVM



4. Abordagens

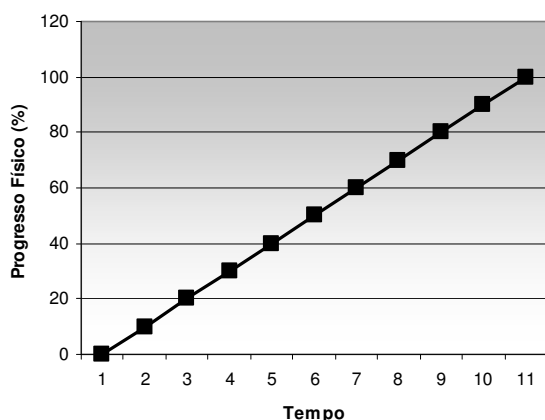
4.1. Abordagem 1: EVM e Relato Ágil (CABRI; GRIFFITHS, 2006)

O trabalho de Cabri e Griffiths (2006) revisa os conceitos de EVM e apresenta considerações de aplicação da técnica em projetos tradicionais e em projetos de desenvolvimento ágil de *software*.

Para os autores, os projetos tradicionais são aqueles que podem ter o escopo bem definido no início do ciclo de vida, poucas mudanças são esperadas e o trabalho pode seguir um plano sequencial de progresso. Essas características contribuem para a aplicação de EVM, já que um dos fatores apontados como fundamentais para a utilização da técnica é ter uma linha de base consistente que permita o confronto com as informações de realização (FLEMING; KOPPELMAN, 2006). Por outro lado, os autores ponderam que no início do ciclo de vida dos projetos de desenvolvimento ágil, o escopo global é definido apenas em alto nível, as mudanças são esperadas durante todo o projeto, levando em consideração que os usuários só entendem o que querem quando vêem alguma parte do produto pronta. Além disso, em projetos ágeis, o trabalho segue um plano iterativo. Estas características tornam os

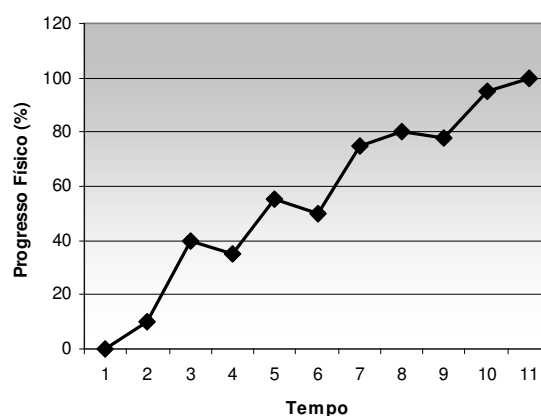
projetos ágeis bem diferentes dos projetos tradicionais. As figuras 2 e 3 destacam o comportamento do progresso físico desses dois tipos de projetos.

Figura 02 – Projeto Tradicional



Fonte: Adaptado de Cabri e Griffiths (2006)

Figura 03 – Projeto Ágil

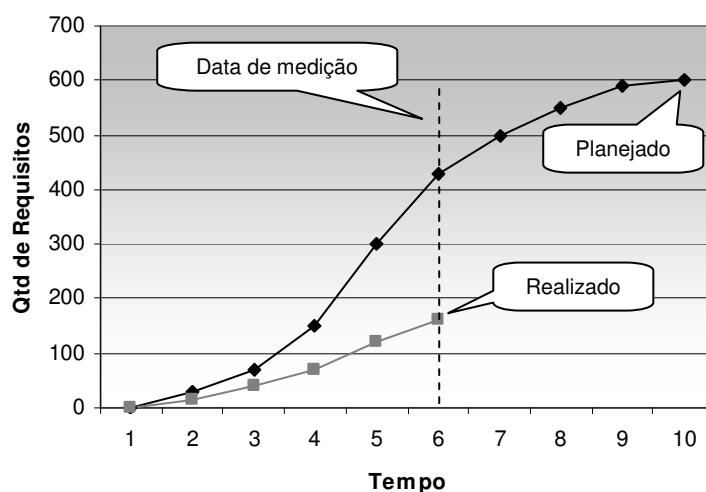


Fonte: Adaptado de Cabri e Griffiths (2006)

Embora os autores afirmem que, em projetos ágeis, o escopo normalmente não seja detalhado totalmente no início do ciclo de vida (por isso, não é possível definir uma linha de base consistente no princípio), afirmam também que as práticas ágeis de controle já instituídas nesses projetos podem ser o caminho mais curto para usufruir dos benefícios sugeridos por EVM. Entre as práticas, destacam os gráficos de controle *Burn-Up* e *Burn-Down*.

Burn-Up é um gráfico que mostra os requisitos acumulados do projeto ao longo do tempo, tendo um aspecto crescente e permitindo a comparação entre a quantidade de requisitos planejados e a quantidade de requisitos realizados (figura 4).

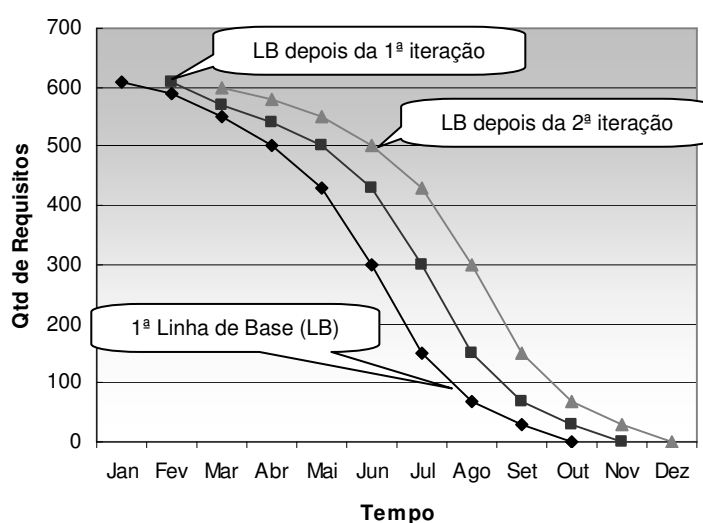
Figura 4 – Burn-UP



Fonte: Adaptado de Cabri e Griffiths (2006)

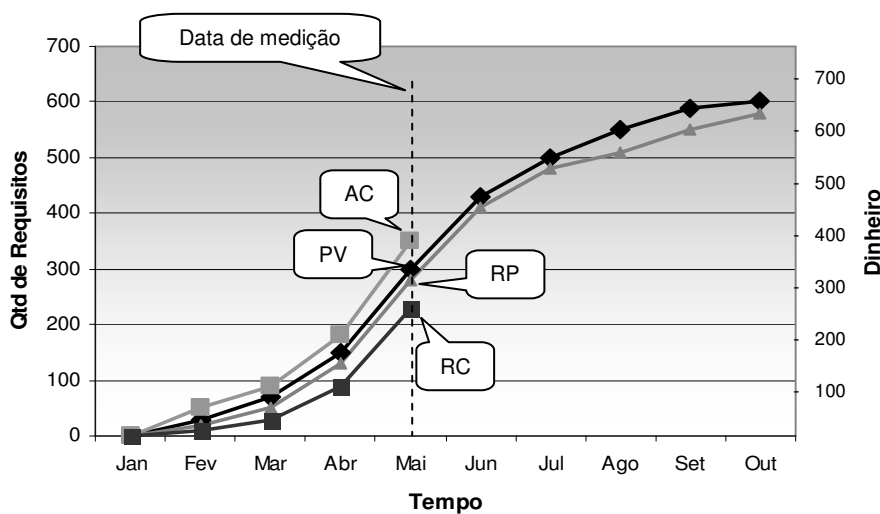
Burn-Down é um gráfico que mostra a quantidade de requisitos que ainda precisam ser feitos e, por isso, tem um aspecto decrescente (figura 5). Além disso, *Burn-Down* pode ser usado para indicar pontos de ruptura que revelam as mudanças de requisitos de uma iteração para outra. Como a cada nova iteração há uma revisão de planejamento, as estimativas são melhoradas à medida que o tempo passa e a linha de base vai ganhando consistência. O gráfico *Burn-Down* normalmente é utilizado para controlar o trabalho por iteração, no entanto, também pode ser utilizado para controlar o trabalho de todo o projeto, como fizeram Cabri e Griffiths (2006) na figura 5.

Figura 5 – *Burn-Down*



Fonte: Adaptado de Cabri e Griffiths (2006)

Cabri e Griffiths (2006) afirmam que se *Earned Value* (EV) ainda não é aplicável diretamente em projetos ágeis seus conceitos certamente já são. Como exemplo, os autores destacam que o *Scheduled Variance* (SV) pode ser facilmente percebido em um gráfico *Burn-Up*, subtraindo a quantidade de requisitos realizados pela quantidade de requisitos planejados (figura 5). A grande diferença é que em um clássico gráfico *Burn* os requisitos não são baseados em uma escala monetária como na abordagem tradicional de EVM. Como consequência, na figura 5, não é possível verificar a eficiência dos gastos do projeto, comparando a quantidade de requisitos realizados contra o *Planned Value* (PV). No entanto, os autores sugerem a utilização de um *Burn-Up* com duas escalas simultâneas: (1) Quantidade de Requisitos e (2) Dinheiro. O *Burn-Up* com duas escalas (figura 06) incorpora os Requisitos Completos (RC) (substituto de *Earned Value*), bem como *Planned Value* (PV) e *Actual Cost* (AC). Desta forma, os três principais elementos de EVM podem ser representados em uma prática já instituída nos projetos ágeis: um gráfico *Burn-Up*.

Figura 6 – Burn-Up com EVM

Fonte: Adaptado de Cabri e Griffiths (2006)

Da figura 6, é possível extrair um índice equivalente ao SPI, dividindo a quantidade de Requisitos Completos (RC) pelos Requisitos Planejados (RP) do período. Similarmente, os autores indicam que um índice equivalente ao CPI pode ser encontrado pela divisão de *Planned Value* (PV) por *Actual Cost* (AC). No entanto, este índice indica apenas a relação entre o dinheiro planejado e o dinheiro gasto, enquanto a abordagem tradicional de EVM para CPI é a divisão entre *Earned Value* (EV) e *Actual Cost* (AC), que revela a eficiência do uso do dinheiro: uma informação diferente.

4.2. Abordagem 2: EVM e SCRUM (SULAIMAN *et al.*, 2006)

Sulaiman *et al.* (2006) apresentam uma abordagem de aplicação de EVM para projetos de *software* integrado ao SCRUM, um modelo de processo ágil de desenvolvimento de *software* (SUTHERLAND *et al.*, 2007). Embora esteja se difundindo muito em projetos de *software*, o SCRUM é um processo aplicável para o desenvolvimento ou gerenciamento de qualquer projeto (SCHWABER, 1996). Entre os principais princípios desse modelo, destacam-se:

- Organização de pequenas equipes de trabalho de modo a maximizar a comunicação, minimizar a supervisão e maximizar o compartilhamento de conhecimento tácito informal;
- Flexibilidade tanto a modificações técnicas quanto de negócios para garantir que o melhor produto possível seja produzido;

- Produção de frequentes incrementos de produto que podem ser inspecionados, ajustados, testados, documentados e expandidos;
- O trabalho de desenvolvimento e o pessoal que o realiza é dividido em participações claras, de baixo acoplamento ou em pacotes;
- Testes e documentação constantes são realizados à medida que o produto é construído;
- O processo fornece a habilidade de declarar o produto concluído sempre que necessário (porque a concorrência acabou de entregar, porque a empresa precisa de dinheiro, porque o cliente precisa das funções ou porque foi para essa data que foi prometido).

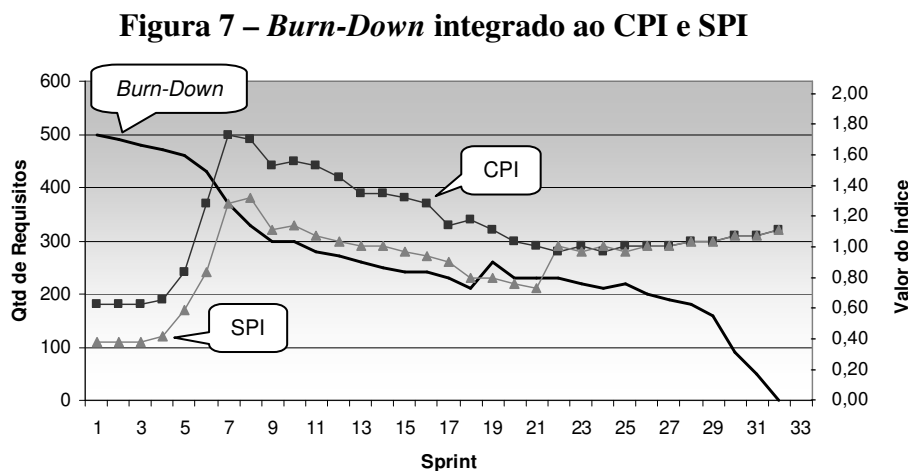
Sulaiman *et al.* (2006) afirmam que SCRUM foca em maximizar o *Return On Investment* (ROI), mas sem definir como gerenciar os custos. Desta maneira, apresentam EVM como técnica capaz de ajudar o cumprimento de tal intento. Os autores adaptaram EVM, usando práticas instituídas em projetos SCRUM e chamaram essa adaptação de AgileEVM. Do SCRUM, foi derivada uma equação para estimar datas de entrega, denominada de *Release Date Velocity* (RDv) e dessa equação foi derivada uma outra equação equivalente para EVM, denominada de *Release Date EV* (RDev). Para os autores, se as equações promoverem resultados semelhantes como instrumentos de estimativa, EVM se torna uma técnica válida em projetos SCRUM.

A primeira equação **RDv** foi definida pela fórmula abaixo, onde **SD** é a data de início de uma *release*, **L** é o tamanho médio das *Sprints* (pequenos ciclos de entrega), **n** é o número de *Sprints*, **w'** é o total de trabalho necessário para fazer uma *release*, e **v** é o trabalho completo por *Sprint*. Desta forma, escreve-se:
$$\mathbf{RDv = SD + L (n + w'/v)}$$
.

A segunda equação **RDev** foi definida pela fórmula abaixo, onde **SD** é a data de início de uma *release*, **L** é o tamanho médio das *Sprints* (pequenos ciclos de entrega), **n** é o número de *Sprints*, **EAC** é a estimativa de custo para o fim do projeto, e **ACn** é o custo real até o momento de medição. Desta forma, escreve-se:
$$\mathbf{RDev = SD + L (n \times EAC/ACn)}$$
.

Para testar a utilidade do AgileEVM, foram utilizados dois projetos. O primeiro foi orçado em meio milhão de dólares, composto por uma equipe que variou entre 5 a 7 membros e com relativa experiência em SCRUM. O primeiro projeto fez uso de *Sprints* semanais e foram relatadas informações mensais de desempenho aos principais envolvidos. O segundo projeto teve a 1ª fase orçada em três milhões de dólares e planejada para durar 18 meses (na época de realização do artigo de Sulaiman *et al.*, o segundo projeto não havia sido totalmente concluído). A equipe esteve composta por 15 a 18 membros e mantinha pouca experiência com SCRUM. No segundo projeto, as *Sprints* foram planejadas com duração de 30 dias.

Os autores afirmam que não existiu distinção entre as projeções das *releases* do AgileEVM nos projetos, o que serviu de validação quanto à expectativa de equivalência entre RDv e RDev. Também foi utilizado um gráfico *Burn-Down* (representação decrescente dos requisitos) integrado aos índices CPI e SPI (figura 7). Para os autores, essa integração demonstra uma importante informação relacionada à eficiência do uso do dinheiro e prazo, que um gráfico *Burn-Down* tradicional não pode evidenciar. Como exemplo, destacam o evidente desempenho insuficiente de prazo, obtido (via SPI) no primeiro projeto entre as *Sprints* 15 e 21 (figura 7).



Fonte: Adaptado de Sulaiman *et al.* (2006)

Sulaiman *et al.* (2006) afirmam que EVM se mostrou útil em projetos SCRUM, pois adicionou valor às práticas instituídas. No entanto, afirmam que, quando não existe a necessidade de controle de custos, a abordagem com EVM não adiciona valor significativo aos projetos. Vale destacar que os membros da equipe do primeiro projeto foram indagados quanto à utilidade do AgileEVM e à possível carga adicional de trabalho por ele promovida. A resposta foi positiva quanto à utilidade e negativa para a eventual carga significativa de trabalho adicional.

4.3. Abordagem 3: EVM e a Incerteza (HANNA, 2009)

O trabalho de Hanna (2009) apresenta e analisa a incerteza do escopo como uma das principais dificuldades para o uso de EVM em projetos de *software*. Para o autor, a aplicação de EVM em projetos modernos de *software* exige ajustes no conceito, pois seu gênesis não previu a volatilidade e a incerteza envolvida nesses projetos (algo tipicamente reconhecido nos projetos ágeis). No entanto, o autor não se limita à apresentação e análise da incerteza envolvida nos projetos de *software*, mas também propõe práticas para tornar o uso de EVM útil e viável mesmo em contextos desse tipo.

Para Hanna a incerteza nos projetos de *software* pode ser dividida em três desafios: realização de protótipos; descoberta de defeitos; e mudanças na arquitetura. A realização de protótipos serve para minimizar a incerteza do escopo, validando conceitos e testando regras de negócio. No entanto, é difícil estimar antecipadamente quantos protótipos e qual o esforço necessário para cada um deles. No que se refere aos defeitos, sabe-se que em projetos de *software* eles existirão. Desta maneira, Hanna afirma que os defeitos podem ser encontrados em tempo de desenvolvimento, em tempo de testes e em tempo de operação. Quanto mais tarde forem descobertos e corrigidos, mais recursos serão consumidos. No que se refere às mudanças na arquitetura, o autor destaca que à medida que se ganha conhecimento e o projeto progride a arquitetura do *software* precisa de ajuste, que normalmente gera trabalho não planejado.

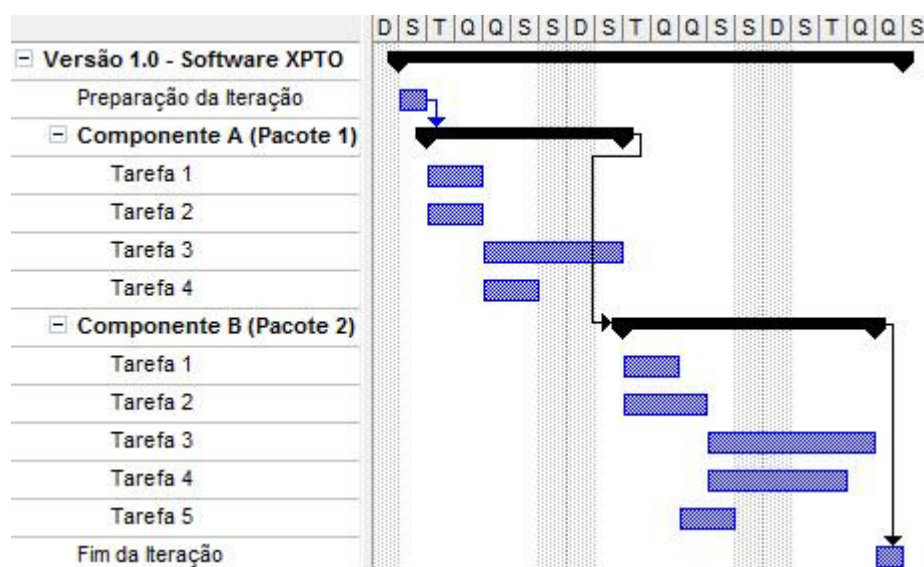
Essas condições podem promover retrabalho (em diferentes graus), que por consequência podem gerar a percepção de progresso negativo nos projetos. Em outras palavras, se um projeto já tinha atingido 50% de avanço físico no segundo mês de execução e no terceiro mês se percebe a necessidade de refazer uma parte importante do escopo (que já havia sido dada como pronta), uma nova referência de conclusão será necessária, exigindo que o valor de avanço real do terceiro mês seja menor do que o valor de avanço real dado no segundo mês. Com isso, tal volatilidade, se não claramente entendida, pode comprometer as expectativas dos envolvidos nos projetos, bem como a utilidade de EVM como instrumento de monitoramento e controle.

Para contornar as dificuldades descritas, o autor propõe o exercício de algumas práticas. Entre elas: (1) decomposição de pacotes de trabalho com duração de duas a quatro semanas; (2) relação de precedência apenas entre pacotes de trabalho e não entre tarefas; (3) elaboração de estimativas através da análise PERT (*Program Evaluation and Review*

Technique), indicando condição pessimista, mais provável e otimista; (4) entregas baseadas em uma *Work Breakdown Structure WBS*; (5) separação dos custos de gerenciamento do trabalho dos custos associados ao trabalho fim, para não promover distorção de análise; e (6) abordagem incremental, entregando *software* frequentemente e atacando as tarefas de maior risco e valor para o cliente.

Vale destacar (para a prática de manter relação de precedência apenas entre pacotes de trabalho) a argumentação do autor de que é desnecessário manter precedências entre as tarefas pequenas dentro de um pacote de trabalho já que elas normalmente são realizadas com pequenas diferenças do planejado, mas que podem se compensar ao longo da duração de todo o pacote. A figura 8 representa essa condição, onde em um pequeno cronograma de iteração existem dois pacotes de trabalho (Componente A e Componente B) planejados sequencialmente. No entanto, as tarefas internas de cada um dos pacotes permanecem sem relacionamento.

Figura 8 – Precedência entre Pacotes



Fonte: Adaptado de Hanna (2009)

Para Hanna, o exercício dessas práticas deve contribuir para os contextos com alto grau de incerteza de escopo, bem como para manter viável o uso e benefícios de EVM como instrumento de monitoramento e controle em projetos de *software*.

4.4. Abordagem 4: Simplificação de EVM para Desenvolvimento Ágil (RUSK, 2009)

O trabalho de Rusk (2009) enfatiza que a simplificação de EVM é um dos principais fatores para a aplicação da técnica em projetos ágeis. No entanto, apresenta que a abordagem ágil também precisa de flexibilidade para que EVM produza benefícios. A simplificação de EVM é apontada como útil principalmente por oferecer um modo de compartilhar os resultados com os envolvidos, especialmente quando os jargões do conceito são desconhecidos.

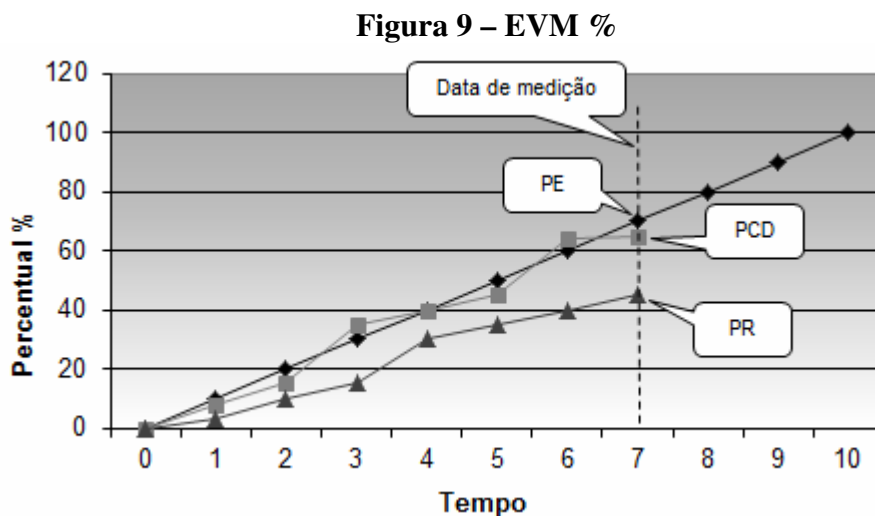
Desta maneira, na visão do autor, é necessário quebrar dois grandes mitos: “agilidade se resume a SCRUM + *eXtreme Programming* (XP)”; e “projetos ágeis não podem definir o escopo em etapas iniciais”. No caso do primeiro mito, Rusk faz questão de lembrar da existência de outros modelos de processos ágeis (*Crystal*; *Feature Driven Development – FDD*, *Dynamic Systems Development Method – DSDM*, etc.), assim como os valores e princípios incluídos no Manifesto Ágil (AGILE ALLIANCE, 2011). Para o segundo mito, Rusk afirma que é bastante comum identificar uma quantidade significativa do escopo no início dos projetos ágeis, citando como exemplo as listas de requisitos do SCRUM (*Product Backlog*) e XP que são criadas no início e mantidas durante todo o projeto.

Com isso, Rusk destaca que a aplicação de EVM precisa apenas de objetivos gerais de escopo, tempo e custo para traçar a expectativa de progresso de um projeto. Em outras palavras, EVM precisa dessas informações em alto nível, podendo naturalmente haver o refinamento ao longo do projeto.

Rusk afirma que algumas práticas dos “agilistas” como o controle de progresso através do gráfico *Burn-Up* mantém equivalência entre os gráficos utilizados em EVM (tipo curva S). No entanto, os “agilistas” têm sua própria terminologia para a referência de andamento dos projetos. O autor se aponta como um “agilista” que percebeu a equivalência de conceitos quando utilizava apenas gráficos *Burn-Up*. Ao monitorar projetos através de *Burn-Up*, Rusk diz que algumas perguntas o despertaram para a equivalência com EVM, tais como: “qual é o andamento do projeto?”; “o progresso é suficiente, mas a custo de quantas horas extras?”; e “se existe atraso no projeto, foi devido a problemas do próprio projeto ou por empréstimo de membros de nossa equipe para outros projetos?”.

Desta maneira, o autor resolveu utilizar EVM em novos projetos e explicou os três conceitos da técnica (*Planned Value*, *Earned Value* e *Actual Cost*) aos membros de sua equipe da seguinte maneira (figura 9): “a partir de agora vamos acompanhar os projetos com três linhas – a 1ª linha mostrará o progresso esperado acumulado (PE) de 0 a 100%; a 2ª linha

mostrará o progresso real (PR); e a 3ª linha mostrará o percentual de consumo do dinheiro (PCD)”.



Fonte: Adaptado de Rusk (2009)

Essa explicação mostra a preocupação em não utilizar os clássicos termos de EVM, apontados pelo autor como de difícil entendimento aos iniciantes. Outro ponto a ser observado na abordagem é que ao invés de apresentar os elementos de EVM em uma perspectiva monetária (caso mais comum), os elementos são plotados em uma perspectiva percentual. No entanto, a eventual necessidade de conversão para a perspectiva monetária exige apenas a multiplicação ou divisão dos elementos (PE, PCD e PR) pelo valor monetário total planejado para o projeto. Isso significa que PE equivale ao *Planned Value*, PCD ao *Actual Cost* e PR ao *Earned Value*.

Rusk encerra seu trabalho afirmando que se for possível definir o escopo em momentos iniciais (mesmo que em alto nível) e estabelecer um plano de progresso para esse escopo (também em alto nível), o uso de EVM se torna viável e útil em projetos ágeis. No entanto, se o projeto for de caráter exploratório e, por isso, não for possível estabelecer objetivos globais para o fim do projeto e o trabalho vislumbrado não passar de poucas semanas, o uso de EVM se torna ineficaz. De qualquer maneira, o autor incentiva a simplificação de EVM tanto para o mundo ágil, como para o mundo clássico da técnica, pois tal prática contribui para o entendimento adequado de elementos fundamentais de monitoramento e controle de projetos.

4.5. Análise comparativa entre as abordagens

Ao estudar as abordagens de uso de EVM em projetos ágeis deste artigo, embora elas sejam independentes, é possível notar algumas semelhanças em suas propostas. As abordagens 1, 2 e 4 sugerem alteração na perspectiva monetária, que é a visão clássica dos conceitos de EVM. No caso das abordagens 1 e 2, a alteração sugerida é a inclusão de uma segunda escala, além da monetária, em gráfico do tipo *Burn* (*Burn-UP* ou *Burn-Down*). No caso da abordagem 4, a alteração sugerida é a conversão da perspectiva monetária em percentual, de modo a representar os elementos básicos (PV, EV e AC) da técnica nessa visão. Entretanto, a abordagem 2 tem como principal característica a incorporação do CPI e SPI em um gráfico *Burn-Down*, que representa o escopo do projeto de forma decrescente. Enquanto isso, a abordagem 1 tem como principal característica a incorporação dos elementos básicos de EVM (PV, EV e AC) em um gráfico *Burn-Up*, que representa o escopo do projeto de forma crescente. A sugestão de duas escalas simultâneas das abordagens 1 e 2 não permite a comparação direta entre todos os elementos plotados nos gráficos, pois não pertencem à mesma perspectiva (ex.: Qtd de requisitos realizados e PV; ou Qtd de requisitos planejados e SPI). A abordagem 2 foi experimentada em dois projetos, enquanto a abordagem 1 não apresentou nenhuma experiência através de projeto, que foi destacado como uma limitação no quadro 4.

Das propostas de alteração na perspectiva monetária de EVM, a abordagem 4 foi considerada a mais versátil e didática, pois não só permite avaliar os elementos básicos da técnica em uma visão percentual ou monetária (multiplicando ou dividindo PV%, EV% e AC% pelo valor orçado do projeto), como também simplifica o vocabulário clássico utilizado.

A abordagem 3, diferente das anteriores, não apresenta nenhuma discussão envolvendo gráficos *Burn-UP*, *Burn-Down* ou gráfico clássico de EVM. Essa abordagem é caracterizada pela sugestão de decompor o escopo do projeto em pacotes de trabalho de 2 a 4 semanas, realizar relacionamentos de precedência apenas em elementos de alto nível (pacotes de trabalho) e estimar esforço com a visão de PERT (otimista, mais provável e pessimista). No entanto, a abordagem 3, assim com a 1 e 4, não registrou experiência real através de projeto.

Com o intuito de facilitar a comparação entre as abordagens, o quadro 4 resume cada uma delas, destacando as características centrais, pontos fortes e limitações percebidas. Embora sejam apresentadas como abordagens independentes, as propostas estudadas não se mostram necessariamente excludentes, permitindo experiências combinadas.

Abordagem	Características Centrais	Pontos Fortes Percebidos	Limitações Percebidas
1) EVM e Relato Ágil (CABRI; GRIFFITHS, 2006).	<ul style="list-style-type: none"> • Incorporação dos três elementos básicos de EVM em um gráfico <i>Burn-Up</i>. • Gráfico <i>Burn-Up</i> com duas escalas. 	<ul style="list-style-type: none"> • Aproveitamento de práticas já instituídas em projetos ágeis para aplicação dos conceitos de EVM (com análise de desvios e projeções). 	<ul style="list-style-type: none"> • Trabalho planejado e realizado não são representados em dinheiro, o que exige 4 elementos básicos para a análise ao invés de 3 da abordagem tradicional de EVM. • Duas escalas diferentes limitam a comparação direta entre os 4 elementos sugeridos, pois 2 elementos pertencem à escala de Requisitos e 2 elementos pertencem à escala Dinheiro. • Não apresentou experiências.
2) EVM e Projetos SCRUM (SULAIMAN <i>et al.</i> , 2006).	<ul style="list-style-type: none"> • Incorporação dos índices CPI e SPI em um gráfico <i>Burn-Down</i>. • Gráfico <i>Burn-Down</i> com duas escalas. 	<ul style="list-style-type: none"> • Aproveitamento de práticas já instituídas em projetos ágeis para aplicação dos conceitos de EVM (com análise de desvios e projeções). 	<ul style="list-style-type: none"> • Duas escalas diferentes limitam a comparação direta entre os 3 elementos sugeridos, pois 2 pertencem à escala dinheiro e 1 pertence à escala Requisitos. • Experiência com apenas dois projetos.
3) EVM e a Incerteza (HANNA, 2009).	<ul style="list-style-type: none"> • Sugestão de práticas relacionadas basicamente à gestão de escopo e tempo (decomposição de pacotes de trabalho com duração de 2 a 4 semanas; precedência apenas entre pacotes de trabalho; estimativas baseadas em PERT). 	<ul style="list-style-type: none"> • Reconhecimento da necessidade de expandir o escopo do projeto durante o ciclo de vida, através de uma WBS evolutiva. • Simplificação do cronograma, estabelecendo relação de precedência entre elementos de alto nível do escopo. 	<ul style="list-style-type: none"> • Não apresenta nenhuma abordagem envolvendo gráficos <i>Burn</i> ou EVM clássico. • Não registra resultados numéricos envolvendo projetos reais.
4) Simplificação de EVM para Desenvolvimento Ágil (RUSK, 2009).	<ul style="list-style-type: none"> • Simplificação da terminologia clássica de EVM. • Apresentação dos elementos básicos na perspectiva percentual. 	<ul style="list-style-type: none"> • Abordagem com boa didática, contribuindo para o entendimento dos iniciantes. • Versatilidade, pois, embora os elementos básicos de EVM sejam apresentados em uma perspectiva percentual, a conversão para a perspectiva monetária exige apenas a multiplicação ou divisão dos elementos básicos pelo valor orçado do projeto. 	<ul style="list-style-type: none"> • Não registra resultados numéricos envolvendo projetos reais.

Quadro 4 – Comparação entre Abordagens

5. Considerações Finais

Este trabalho apresentou quatro abordagens de aplicação de EVM em projetos ágeis de *software*. Em cada uma delas foram ressaltadas características particulares. No entanto, a razão de existirem representa o desejo de adaptar conceitos originais de EVM para a aplicação mais alinhada ao contexto atual de produção de *software*.

Embora não tenha sido observado consenso absoluto sobre quais aspectos de EVM precisam ser adaptados para tornar a técnica mais aderente aos projetos ágeis de *software*, notam-se pontos comuns que as abordagens estudadas tentam solucionar. Entre eles, a necessidade de definir uma linha de base e medir progresso em projetos ágeis de *software* de forma consistente, levando em consideração a natureza intangível e a dificuldade de definição de escopo. Além disso, reconhecer e controlar com maior flexibilidade as mudanças no escopo de projetos de *software* são apontados como fatores fundamentais nessas abordagens. Desta maneira, para o amadurecimento das propostas apresentadas e o surgimento de outras, é necessário que mais experiências com projetos reais sejam feitas combinando práticas ágeis e EVM.

6. Referências Bibliográficas

AGILE ALLIANCE. Manifesto for agile software development. Disponível em <<http://www.agilemanifesto.org>>. Acesso em abril, 2011.

ANBARI, Frank T. Earned Value Project Management: Method and Extensions. **Project Management Journal**. v.34, n.34, p.12-23, Washington, USA, December, 2003.

BOOCH, Grady. Melhores Práticas do Desenvolvimento de Software. In: KRUCHTEN, Philippe. **Introdução ao RUP – Rational Unified Process**. Addison-Wesley/Ciência Moderna, 2003. p.3-13.

CABRI, Anthony; GRIFFITHS, Mike. Earned Value and Agile Reporting. In: PROCEEDINGS OF THE CONFERENCE ON AGILE 2006, p.17-22, 23-28 July, 2006.

COCKBURN, Alistair. **Agile Software Development**. Addison-Wesley, 2002.

FLEMING, Quentin W.; KOPPELMAN, Joel M. Start With “Simple” Earned Value On All Your Projects. **CrossTalk: The Journal of Defense Software Engineering**, June, 2006.

FLEMING, Quentin W.; KOPPELMAN, Joel M. Earned Value Management: A Powerful Tool for Software Projects. **CrossTalk: The Journal of Defense Software Engineering**, July, 1998.

HANNA, Robert A. Earned Value Management Software Projects. In: 3th IEEE INTERNATIONAL CONFERENCE ON SPACE MISSION CHALLENGES FOR INFORMATION TECHNOLOGY, 2009.

JONES, Carpes. Social and Technical Reasons Software Project Failures. **CrossTalk: The Journal of Defense Software Engineering**, June, 2006.

KIM, EunHong.; WELLS JR., William G.; DUFFEY, Michael R. A model for effective implementation of Earned Value Management methodology. **International Journal of Project Management**, v. 21, n. 5, p.375-382, 2003.

LACERDA, Isac M. **Fatores Críticos de Sucesso no Uso de Earned Value Management em Projetos de Desenvolvimento de Software e a Relação com a Qualidade Percebida**. 2009, 200f. Dissertação (Mestrado em Sistemas de Gestão). Universidade Federal Fluminense, Niterói, 2009.

LI, Jinhua; MA, Zhibing; DONG, Huanzhen. Monitoring Software Projects With Earned Value Analysis and Use Case Point. In: PROCEEDINGS OF THE 7th IEEE/ACIS INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION SCIENCE, p.475-480, 14-16 May, 2008.

MAQSOOD, Manzil; JAVED, Talha. Practicum in Software Project Management – An Endeavor to Effective and Pragmatic Software Project Management Education. In: PROCEEDINGS OF THE 6th JOINT MEETING OF THE EUROPEAN SOFTWARE ENGINEERING CONFERENCE. Session: Widened Software Engineering, p.471-480, Dubrovnik, Croatia, 2007.

NORONHA, Daisy P.; FERREIRA, Sueli. M. S. P. Revisões da Literatura. In: CAMPELLO, Bernadete S., CENDÓN, Beatriz V.; KREMER, Jeannette M. K. **Fontes de Informação para Pesquisadores e Profissionais**. Editora UFMG, p.191-198, 2000.

PRESSMAN, Roger S. **Engenharia de Software**. 6nd. Mc Graw Hill – São Paulo, 2006.

PUTZ, Peter; MALUF, David A.; BELL, David G.; GURRAM, Mohana M.; HSU, Jennifer; PATEL, Hemil N.; SWANSON Keith J. In: Earned Value Management at NASA: An Integrated, Lightweight Solution. IEEE Aerospace Conference, Montana, 2007.

RUSK, John. Earned Value for Agile Development. **DoD Software Tech News (Journal of Software Technology)**, v. 12, n.3, September, 2009.

SCHWABER, Ken. Controlled-Chaos: Living on the Edge. **American Programmer**, April, 1996.

SULAIMAN, Tamara; BARTON, Brent; BLACKBURN, Thomas. AgileEVM – Earned Value Management in SCRUM Projects. In: PROCEEDINGS OF CONFERENCE ON AGILE 2006, p.7-16, July 23-28 – Minnesota, 2006.

SUTHERLAND, Jeff; VIKTOROV, Anton; BLOUNT, Jack; KIKOLAI, Puntikov. Distributed SCRUM: Agile Project Management With Outsourced Development Teams. In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, Hawaii, 2007.

VARGAS, Ricardo V. Earned Value Analysis in the Control of Projects: Success or Failure? In: 47th ANNUAL MEETING OF AACE INTERNATIONAL - Orlando, USA, 2003.