



PUC

ISSN 0103-9741

Monografias em Ciência da Computação
nº 18/10

Análise de Riscos em Projetos: uma abordagem por reflexão computacional baseada em lógica

Dárlinton Barbosa Feres Carvalho

Edward Hermann Haeusler

Clarisse Sieckenius de Souza

Carlos José Pereira de Lucena

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22453-900

RIO DE JANEIRO - BRASIL

Análise de Riscos em Projetos: uma abordagem por reflexão computacional baseada em lógica ¹

Dárlinton Barbosa Feres Carvalho, Edward Hermann Haeusler, Clarisse Sieckenius de Souza, Carlos José Pereira de Lucena

{dcarvalho,hermann,clarisse,lucena}@inf.puc-rio.br

Resumo. Gerenciamento de risco é uma disciplina de gerência de projetos que possui como meta identificar, analisar e tratar fatores de risco, e com isso aumentar a chance de sucesso de seus projetos. Mesmo com objetivos tão nobres, esta é a disciplina com menor grau de maturidade nas empresas. 75% dos gerentes de projeto não praticam qualquer forma detalhada de gerência de riscos e entendem apenas vagamente os conceitos relacionados a essa disciplina, bem como suas consequências. Para auxiliar os gerentes de projetos a analisar os fatores de risco em um projeto, é proposto neste trabalho um modelo para reflexão computacional baseado em lógica. A epistemologia utilizada na construção do modelo, projetado para ser verificado por um verificador de modelos baseado em lógica, é apresentada junto a um exemplo. O modelo proposto é avaliado por entrevistas e experimentos com usuários potenciais do método.

Palavras-chave: Análise de Riscos, Categorização de Fatores Chave de Risco, Verificador de Modelos, Reflexão Computacional.

Abstract. Risk management is a discipline of project management that aims to identify, analyze and handle project risks, to increase the probability of success. Even with so noble objectives in the project management field, this area has the lowest maturity level in companies. 75% of the project managers do not apply any kind of systematic risk management tactics and vaguely understand the concepts related to this discipline, not to mention the consequences. This work proposes a tool to help project managers to do risk analysis. This tool is a model based on logic used for computational reflection about risks in projects. The epistemology used to build this model, taking advantage of a logic-based model checker to verify its properties, is presented together with an example. Potential users for this methodology evaluate the proposed model with interview and experiment.

Keywords: Risk Analysis, Categorizing Key Drivers of Risk, Model Checker, Computational Reflection.

¹Trabalho patrocinado pelo Ministério de Ciência e Tecnologia da Presidência da República Federativa do Brasil através da Bolsa de Doutorado CNPq 142620/2009-2.

In charge of publications:

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22451-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3527-1516 Fax: +55 21 3527-1530
E-mail: bib-di@inf.puc-rio.br
Web site: <http://bib-di.inf.puc-rio.br/techreports/>

Sumário

1	Introdução	1
2	Metodologia	2
2.1	Análise de Risco	2
2.2	Verificador de Modelos	6
2.3	Proposta	6
3	Avaliação	11
4	Conclusão	13
	References	13
A	Termo de Consentimento	15
B	Formulário da entrevista	17
C	Lâminas utilizadas para explicar a nova metodologia	23
D	<i>Checklist</i> de fatores de risco chave utilizada no experimento	28

1 Introdução

Gerenciamento de risco é uma disciplina de gerência de projetos que possui como meta identificar, analisar e tratar fatores de risco (Committee 2008). Para realizar esta tarefa, os gerentes de projeto utilizam um conjunto de princípios e práticas para aumentar a chance de sucesso de seus projetos. Mesmo possuindo objetivos tão nobres na gestão de projetos, esta é a disciplina com menor grau de maturidade nas empresas, sendo que 75% dos gerentes de projeto não praticam qualquer forma detalhada de gerência de riscos e entendem apenas vagamente os conceitos relacionados a essa disciplina, bem como suas consequências (Bakker, Boonstra & Wortmann 2009).

Na literatura, encontram-se diversas metodologias para realizar análise de risco. Os artigos são bem diversos e vão de uma simples *checklist* para avaliações de projetos em um minuto (Tiwana & Keil 2004), a *frameworks* para criação de métodos adaptados a cada projeto (Keil, Cule, Lyytinen & Schmidt 1998, Varnell-Sarjeant 2008, Warkentin, Moore, Bekkering & Johnston 2009, Alberts & Dorofee 2009). Também se encontra trabalhos que buscam padrões em documentação de projetos para serem generalizados e utilizados na criação de novos métodos (Wallace, Keil & Rai 2004), ou até mesmo trabalhos mais recentes discutindo a forma e a eficácia dos métodos para análise de risco (Bannerman 2008, Bakker et al. 2009). Como a literatura é extensa, neste trabalho são considerados principalmente artigos relacionados com o gerenciamento de projetos para desenvolvimento de software.

Com o objetivo de auxiliar os gerentes de projeto a realizar análise de risco, é proposto neste trabalho um *framework* analítico baseado em lógica, através da especificação formal de um modelo para análise de risco. Para isso, é apresentada a epistemologia utilizada na construção desse modelo, que é composta principalmente do conceito de risco, visão de risco sistêmico, a utilização de fatores chave de risco, categorias desses fatores, a relação entre as categorias e o sistema de qualificação desses fatores. A proposta deste trabalho é uma extensão do *framework* para a categorização de fatores chave de risco criado por Alberts & Dorofee (Alberts & Dorofee 2009), com a definição de um modelo analítico de fatores de risco em um projeto.

A análise de risco é realizada através de um modelo com os fatores de riscos e as relações lógicas entre eles. As relações entre os fatores de riscos permite criar um sistema em que um risco influencia diretamente em outro, e por consequência de sucessivas influências pode afetar o resultado global do projeto. A verificação do modelo é realizada através da enumeração exaustiva de todas as possibilidades de interação definidas entre os elementos que compõe o modelo. A ferramenta utilizada para realizar essa tarefa é chamada de verificador de modelos. O usuário pode especificar assertivas lógicas e utilizar o verificador de modelos para certificar-se de propriedades do projeto, sendo também possível realizar alterações no modelo para simular outros cenários hipotéticos.

O modelo para análise de risco é avaliado através de entrevistas com usuários potenciais do método proposto. Cada usuário utilizou o método para analisar os riscos de um projeto de seu interesse. Ao final do experimento, avaliou-se a pontos fortes, fracos e possíveis melhorias. Quase todos os entrevistados disseram que utilizariam o método proposto em seus projetos, mas com as devidas adaptações a realidade de seus projetos.

O restante deste artigo está organizado como segue. Na Seção 2, detalha-se a metodologia proposta para realizar análise de risco em projetos. A Seção 3 apresenta uma avaliação do método proposto realizada por experimento e entrevista com usuários. Por fim, o artigo é concluído na Seção 4.

2 Metodologia

Para realizar análise de risco de projetos, é proposto um *framework* analítico através de uma especificação formal baseada em lógica dos fatores de risco de um projeto. Esse modelo é uma extensão de uma metodologia definida para análise de risco em projetos de software, apresentada na Seção 2.1. A extensão é realizada através da criação de uma especificação formal que pode ter propriedades validadas através de um verificador de modelos, que é apresentado na Seção 2.2. Com este novo método é possível a avaliação das consequências de um fator de risco em outro, bem como a simulação de diversos cenários hipotéticos para análise de risco. Na Seção 2.3 está a especificação do modelo proposto em uma linguagem formal, com as relações entre as entidades do modelo.

2.1 Análise de Risco

Gerenciamento de risco é uma disciplina de gerência de projetos que possui como meta identificar, analisar e tratar fatores de risco. Para isto, é utilizado um conjunto de princípios e práticas para aumentar a chance de se ter como resultado um projeto bem sucedido. Esta é a disciplina com menor grau de maturidade nas empresas das quatro áreas definidas pelo Instituto de Gerenciamento de Projetos (Committee 2008). 75% dos gerentes de projeto não praticam qualquer forma detalhada de gerência de riscos e entendem apenas vagamente os conceitos relacionados a essa disciplina, bem como suas consequências (Bakker et al. 2009).

Para responder o questionamento de que se realizar análise de risco em projetos de software aumenta de fato o sucesso dos projetos, Bakker *et al.* (Bakker et al. 2009) realizaram uma revisão da literatura. O primeiro problema atacado nesta pesquisa foi definir o que é o sucesso na execução de um projeto de software. A definição clássica de sucesso em projetos consiste em uma simples comparação entre o que foi planejado, considerando cronograma, orçamento e requisitos, e o que foi entregue. Considerando esta avaliação, estima-se que pouco mais de 10% de todos os projetos são considerados bem sucedidos. A pesquisa também descobriu que projetos com a utilização de métodos para a análise de risco que consideram a participação de diversas pessoas (*stakeholders*) no processo, conseguiram aumentar a percepção de sucesso. Percepção de sucesso é uma medida em que os participantes do projeto dizem se consideram o projeto bem sucedido ou não. Pequenos desvios das metas planejadas são compensados pelo conhecimento das dificuldades enfrentadas, e mesmo que não completamente satisfeitas o projeto é considerado um sucesso. Na literatura é reportado que cerca de 55% de projetos passaram por dificuldades, tiveram que se adaptar, e finalizaram com uma avaliação de sucesso. Portanto, a realização de análise de risco colabora para o sucesso no desenvolvimento de um projeto, e a participação dos envolvidos no projeto (*stakeholders*) nessa análise aumenta a percepção de sucesso do projeto como um todo.

Na literatura, encontram-se diversas maneiras para realizar análise de risco. É possível agrupar essas diversas formas em quatro classes (Bannerman 2008): *checklists*, *frameworks* analíticos, modelos de processo, estratégias de resposta a risco. Listas dos maiores fatores de risco ou para sucesso em projetos de software são os mais comuns na literatura e na prática. Essas listas geralmente são construídas a partir da experiência de participantes em projetos e do estudo sobre histórico de projetos. Um exemplo dessas *checklist* é a clássica avaliação de qualquer projeto proposta a ser realizada em apenas um minuto (Tiwana

& Keil 2004). Modelos para moldar o raciocínio de avaliadores sobre a análise de risco são conhecidos como *frameworks* analíticos ou modelos de processo. Ambos são muito parecidos e baseados em *checklists*, mas possuem um modelo mais elaborado para agrupar fatores de risco e lidar com eles. A diferença entre os *frameworks* analíticos e os modelos de processos está no fato dos primeiros serem apenas molduras para o raciocínio dos avaliadores e os outros condicionarem a análise em um processo com passos bem definidos para realizar análise de risco. A última categoria é sobre os outros métodos de gerenciamento do risco, que são voltados para estratégias de resposta a risco. Essas estratégias podem ser divididas em quatro categorias: estratégias que evitam o risco (prevê alterações no projeto diante certas situações), transferência de risco (em situações de alto risco transfere a responsabilidade do projeto a gerências mais competentes a resolver os problemas), eliminação por antecipação (busca identificar sinais de problema e adotar práticas para evitar que se tornem uma ameaça real) e aceitação (define uma série de ações passivas ou ativas em resposta a eventos de risco).

Neste trabalho, propõe-se um *framework* analítico através da especificação formal de um modelo para análise de risco. A seguir, é apresentada a epistemologia utilizada na construção desse modelo, que é composta principalmente do conceito de risco, a visão de risco sistêmico, a utilização de fatores chave de risco, as categorias desses fatores, a relação entre as categorias e o sistema de qualificação desses fatores chave.

Da teoria clássica de decisão, risco é definido por três variáveis relacionadas segundo apresentado pela fórmula $R = P \times I$. Nessa fórmula, R é a exposição de risco atribuída a um fator de risco específico. P é a probabilidade de um evento relacionado ao fator de risco se realizar. I é o impacto, ou magnitude do evento. Na teoria clássica, o impacto pode ser positivo ou negativo, sendo que uma visão mais moderna desta definição entende que se o impacto for positivo então se trata de uma oportunidade e não de um risco, portanto deve ser tratado pela gerência de oportunidades (Committee 2008, Bannerman 2008). Neste trabalho, consideramos apenas impactos negativos, que representam ameaças a projetos.

Em um projeto real, o número de ameaças é elevado para ser analisado manualmente pelos gerentes de projetos. Esta limitação no processamento levou os gerentes de projetos a adaptarem a metodologia clássica (Bannerman 2008). A primeira adaptação é que ao invés de definir o gerenciamento dos riscos a partir de uma ordenação pelo valor de R , os gerentes se importam mais com a magnitude da perda potencial, mesmo que com baixa probabilidade. Para os gerentes, I é mais importante do que P . Esta avaliação também deve estar relacionada ao fato de que eles preferem caracterizar verbalmente um risco a sua representação probabilística, porque eles, em geral, são céticos sobre a redução da larga dimensionalidade do risco a um único número. Por fim, os gerentes tendem a não aceitar estimativas de risco dadas a eles, pois eles veem o risco como objeto a ser controlado. Gerenciar riscos faz parte das funções do gerente de projeto. Só que estas adaptações levam os gerentes a ignorarem um grande número de ameaças, mesmo as com alta probabilidade de acontecer, já que na prática eles conseguem gerenciar de 3 a 5 fatores de riscos, geralmente escolhidos pela magnitude do impacto mesmo com baixa probabilidade de acontecer.

Para melhorar a representação dos fatores de riscos, é utilizada a metodologia proposta em (Alberts & Dorofee 2009). Nesta metodologia, o risco é entendido como uma relação de causa e efeito expressa pela probabilidade de acontecer uma ameaça que leva a uma consequência, que por sua vez possui um impacto no projeto. Há diversos riscos relacio-

nados a um mesmo fator de risco. Para lidar com o grande número de riscos associados ao projeto como um todo, é escolhido apenas alguns fatores chave de riscos para agrupar os riscos individuais. Neste modelo, o impacto de um risco influencia diretamente seu fator chave associado, mas também pode influenciar outros fatores chave de risco, ou seja, é um modelo que possui uma visão sistêmica do risco. Para isso, é utilizada uma definição de risco em que um evento potencial gera consequências sob determinadas condições. Essas consequências por sua vez podem ser parte de condições do fator de risco chave ao qual o risco está associado, ou até mesmo ser um evento potencial de outro fator de risco. Então, um fator chave de risco possui diversas condições que favorecem positivamente ou negativamente para ocorrência de um evento futuro. Um fator de risco chave pode influenciar outros fatores de risco chave.

Um projeto possui diversos objetivos, sendo que cada objetivo possui fatores de risco, que por sua vez possuem condições e eventos tanto positivos quanto negativos. Para facilitar a organização nas definições das relações entre os fatores de risco, são definidas seis categorias dos fatores de risco. As categorias, bem como as relações de influência entre elas são apresentadas na Figura 1. As categorias são representadas pelas caixas, e as setas entre as caixas indicam a influência entre as categorias. Por exemplo, um fator de risco da categoria Objetivos (*Objectives*) pode exercer influência sobre fatores de risco das categorias Preparação (*Preparation*) e Resiliência (*Resilience*). Uma definição completa das categorias está disponível em (Alberts & Dorofee 2009).

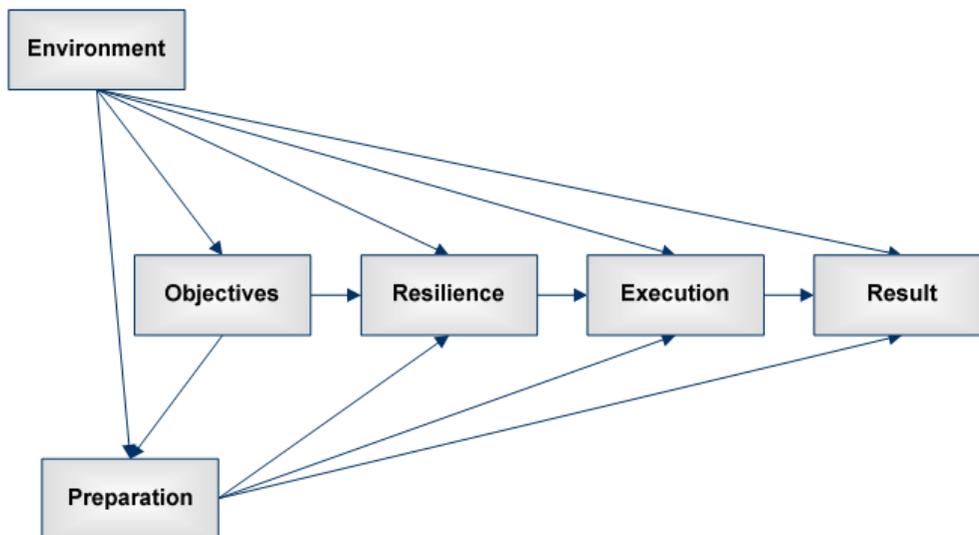


Figura 1: Relação entre as categorias dos fatores de risco

Para analisar um projeto, portanto, devem-se analisar os fatores chave de risco, relacionados com os objetivos do projeto. A avaliação de um fator de risco é realizada através de uma qualificação em escala de *Likert*, que está relacionada com a probabilidade em se alcançar sucesso ou insucesso, e as razões (*rationale*) da escolha descritas textualmente.

Um exemplo de resposta a um fator de risco é apresentado na Figura 2. O fator chave de risco avaliado é o processo utilizado no desenvolvimento do projeto. Apesar de não estar explícito na figura, vale dizer que este fator chave de risco pertencente à categoria Preparação. A pergunta direciona o avaliador a responder Sim (*Yes*), se as condições

relacionadas a este fator chave de risco sinalizarem baixíssimo risco, seguindo um contínuo até Não (*No*), que representa risco altíssimo. Nas razões (*Rationale*) o avaliador descreve textual o que foi considerado na avaliação, sinalizando os elementos que contribuíram positivamente, e os que contribuíram negativamente.

Question	Answer				
3. Is the process being used to develop and deploy the system sufficient? Consider: process design; measurements and controls; process efficiency and effectiveness; acquisition and development life cycles; training	No	Likely no	Equally likely	Likely yes	Yes
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<p>Rationale</p> <ul style="list-style-type: none"> + Previous programs have a 90% history of delivering on-time. - The process for integration testing is likely inadequate. Historically, integration testing has used "verbal" agreements between a few managers who already know each other. With this system, there are managers and team leads who have never worked together and there are other barriers in place that make "verbal" agreements tenuous. - There are a lot of brand new programmers (45%). - This program required a significant change in our standard processes. There was no new training created for the new processes. - QA did not have a chance to review the new and revised processes before they were put into practice. - The person who developed the new processes quit last week. 					

Figura 2: Exemplo de avaliação de um fator de risco

Na metodologia apresentada em (Alberts & Dorofee 2009), não é proposta nenhum método para análise de risco, mas são apresentadas diversas possibilidades para analisar os fatores de risco. A análise mais trivial seria montar um gráfico relacionando todos os fatores de risco e suas avaliações. Variando um pouco mais este gráfico, podem-se agrupar os fatores de risco de acordo com suas categorias, mas não é apresentada nenhuma forma de simular configurações alternativas da avaliação dos fatores de risco, ou do impacto de um fator de risco em outro.

2.2 Verificador de Modelos

A análise de risco é realizada através da verificação de um modelo, que é definido a partir dos fatores de riscos e as relações lógicas entre eles. A verificação de modelos é realizada através da enumeração exaustiva de todos os estados possíveis. A ferramenta utilizada para realizar essa verificação de modelos é chamada de Verificador de Modelos.

Verificação de modelos é um método para verificar formalmente sistemas concorrentes modelados por estados finitos (Clarke, Emerson & Sistla 1986). Uma especificação sobre o sistema é expressa por fórmulas de lógica temporal, e eficientes algoritmos simbólicos são usado para percorrer o modelo definido pelo sistema e verificar se a especificação é consistente ou não. A verificação de modelos, em que se verifica automaticamente a validade de propriedades, utilizada é baseada em sistemas reativos. Sistemas reativos são definidos basicamente por estados e transições. Estado é a descrição do sistema em um instante de tempo, ou seja, os valores associados as suas variáveis naquele instante. Já uma transição é a relação entre dois estados. A computação é representada por uma sequencia infinita de estados, sendo que cada estado é obtido a partir de um estado anterior através de uma transição entre eles.

Vale ressaltar que mesmo utilizando verificação de modelos não há como garantir a equivalência dos modelos com a realidade do projeto, mas já se consegue verificar propriedades relacionadas aos fatores de riscos que estejam especificadas no modelo. As relações entre os fatores de riscos permite criar um sistema em que um risco influencia diretamente outro risco, e por consequência de sucessivas influências pode afetar o resultado global do projeto. Dessa forma, possibilita-se descobrir consequências relevantes sobre os cenários avaliados para os projetos, ou mesmo ter garantias em relação ao que se está modelado.

Existem duas abordagens para implementar verificação de modelos: a abordagem lógica e a abordagem baseada em teoria dos autômatos. Neste trabalho, utiliza-se a abordagem lógica e o verificador de modelos *Symbolic Model Verifier* (SMV) (McMillan 1992). Nesta abordagem, um sistema reativo é descrito através de um tipo de grafo de transição de estados, chamado de *estrutura de Kripke*. A ferramenta SMV possui uma linguagem para especificação de modelos que é utilizada para descrever os sistemas de estados finitos. Essa linguagem possui sintaxe baseada em lógica temporal do tipo *Computation Tree Logic* (CTL) (Emerson 1990) para expressar as propriedades a serem verificadas. O sistema SMV está disponível para download na Internet².

O usuário pode especificar assertivas lógicas e utilizar o verificador de modelos para certificar-se de propriedades do projeto, sendo também possível realizar alterações no modelo para simular outros cenários. Um exemplo de propriedade que pode ser avaliada é se algum fator de risco (variável) pode alcançar um determinado nível de risco (valor). Variações nos valores iniciais das variáveis (avaliação dos fatores de risco) e nas relações ente elas possibilitam criar diferentes cenários a serem verificados.

Na próxima seção, são apresentados detalhes da especificação do modelo e as possibilidades de adaptação deste método para projetos específicos.

2.3 Proposta

As respostas dadas pelos gerentes de projetos a uma *checklist* de fatores de risco são utilizadas para analisar os riscos de um projeto. Os elementos principais nesta análise são

²<http://www.cs.cmu.edu/~modelcheck/smv.html>

as avaliações dos fatores chave de risco e as relações entre eles. Os riscos potenciais do projeto são anotados textualmente e agrupados em seus respectivos fatores chave de risco. A avaliação sistêmica de todos os fatores chave de riscos, bem como dos relacionamentos entre eles, determina uma análise de risco do projeto como um todo. Essa análise é realizada através da verificação de um modelo de influências de fatores chave de riscos conforme apresentado a seguir.

É proposto um modelo que descreve um sistema em que um fator chave de risco pode influenciar na avaliação de outro fator, tendo essa relação definida pela categoria do fator chave de risco. Os riscos são descritos como eventos que causam uma mudança da avaliação de risco em um fator. Portanto, os riscos de um fator chave de risco são os eventos que causam mudança na avaliação do risco. A partir de uma avaliação inicial, a ocorrência do evento (risco) leva a avaliação para outro valor (final). Um risco pode afetar um fator chave de risco, que por sua vez pode afetar outros fatores de risco. A relação entre os fatores de risco é determinada pelas categorias, lembrando que cada fator de risco pertence a uma categoria.

Para exemplificar a relação entre os fatores de risco, é apresentado na Figura 3 um conjunto de 10 fatores de risco e o relacionamento entre eles, conforme sugestão de fatores definido pelo framework (Alberts & Dorofee 2009). Os fatores chave de risco são representados por retângulos nomeados e a categoria é apresentada entre parênteses após o nome do fator de risco. As setas indicam que a avaliação de um fator de risco influencia na avaliação do fator de risco na outra ponta, no sentido da seta.

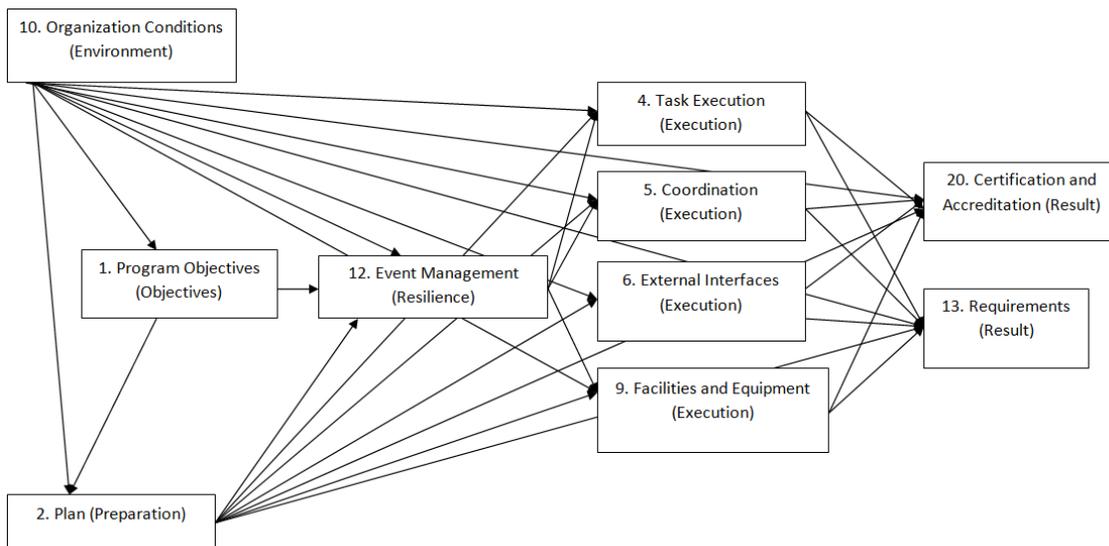


Figura 3: Modelo do sistema de influência entre os fatores chave de risco

O modelo foi implementado na linguagem do verificador de modelos **SMV**. Dessa forma, o modelo proposto suporta a ocorrência de uma série infinita de eventos e de influências entre os fatores de risco. Propriedades lógicas definidas em **TCL** são verificadas pelo **SMV**. A avaliação dos fatores de risco considerada é de três níveis de risco: alto, médio e baixo. A seguir são apresentados alguns detalhes de implementação.

Cada fator chave de risco é programado como um módulo que possui como variáveis

o estado, que representa a avaliação desse fator, e os eventos (riscos) que influenciam na avaliação. Cada módulo também define as transições da avaliação do fator de risco, de acordo com os eventos definidos pelo usuário. A seguir é apresentado o código da implementação de um fator de risco com três eventos do usuário.

```
MODULE d10_environment_organization_conditions
VAR
  state : {h,m,l};
  -- user events
  ev1 : {1,0};
  ev2 : {1,0};
  ev3 : {1,0};
ASSIGN
  next(state) := case
    -- user events
    state = m & ev1 : h;
    state = m & ev2 : h;
    state = m & ev3 : l;
    1 : state;
  esac;
```

Os fatores chave de risco que podem ser influenciados por outros fatores são definidos como módulos parametrizados. A influência é programada como uma transição, notificada pelo parâmetro. No modelo proposto, um fator de risco avaliado como alto pode fazer com que a avaliação de outros fatores de riscos sob sua influência também sejam avaliados com risco alto. Também é definido que se todos os fatores chave de risco que influenciam um determinado fator são avaliados com risco médio então esse fator é avaliado como risco alto. Outra transição definida é que se o fator chave de risco tiver avaliação baixa e algum dos fatores de risco que influenciam esse fator tiver com nível médio então a avaliação aumenta para médio.

```
MODULE d20_result_certification_and_accreditation (d10_env_state,d2_prep_state,
  d4_exec_state,d5_exec_state,d6_exec_state,d9_exec_state)
VAR
  state : {h,m,l};
  -- user events
  ev1 : {1,0};
  ev2 : {1,0};
  ev3 : {1,0};
ASSIGN
  next(state) := case
    -- user events
    state = l & ev1 : m;
    state = l & ev2 : h;
    state = l & ev3 : m;
    -- transitions from relationships among the driver categories
    d10_env_state = h : h;
```

```

    d2_prep_state = h : h;
    d4_exec_state = h : h;
    d5_exec_state = h : h;
    d6_exec_state = h : h;
    d9_exec_state = h : h;
    d10_env_state = m & d2_prep_state = m & d4_exec_state = m
        & d5_exec_state = m & d6_exec_state = m & d9_exec_state = m : h;
    state = 1 & (d10_env_state = m | d2_prep_state = m | d4_exec_state = m
        | d5_exec_state = m | d6_exec_state = m | d9_exec_state = m) : m;
    1 : state;
esac;

```

O risco global do projeto é definido a partir da avaliação dos fatores de risco chave da categoria resultado (*result*). Se algum fator de risco da categoria resultado tiver avaliação de alto nível de risco então o resultado global é algo. Se os fatores de risco que influenciam o resultado global do projeto tiverem avaliação de nível médio então o resultado pode ser médio ou alto. Vale lembrar, que o modelo deve ser adaptado a realidade de cada projeto a ser avaliado e aqui é apresentado apenas um exemplo de modelo.

```

MODULE global_risk (d13_result_state,d20_result_state)
VAR
    state : {h,m,l};
ASSIGN
    next(state) := case
        -- transitions from relationships among the driver categories
        d13_result_state = h | d20_result_state = h : h;
        d13_result_state = m & d20_result_state = m : {m,h};
        1 : state;
    esac;

```

O módulo principal do modelo consiste da definição de uma variável para cada fator de risco. Nestas definições, o relacionamento entre os fatores de risco é estabelecido pelos parâmetros dos módulos.

```

MODULE main
VAR
    d10 : process d10_environment_organization_conditions;
    d1  : process d1_objectives_program_objectives (d10.state);
    d2  : process d2_preparation_plan (d10.state,d1.state);
    d12 : process d12_resilience_event_management (d10.state,
        d1.state,d2.state);
    d4  : process d4_execution_task_execution (d10.state,d2.state,d12.state);
    d5  : process d5_execution_coordination (d10.state,d2.state,d12.state);
    d6  : process d6_execution_external_interfaces (d10.state,
        d2.state,d12.state);
    d9  : process d9_execution_facilities_and_equipment (d10.state,
        d2.state,d12.state);

```

```

d13 : process d13_result_requirements (d10.state,d2.state,d4.state,
      d5.state,d6.state,d9.state);
d20 : process d20_result_certification_and_accreditation (d10.state,
      d2.state,d4.state,d5.state,d6.state,d9.state);
g    : process global_risk   (d13.state,d20.state);

```

O estado inicial de cada módulo é definido conforme a avaliação de cada fator de risco na *checklist*. Caso algum fator de risco não tenha resposta na *checklist*, o modelo suporta a definição de todos os níveis para serem considerados na verificação do modelo. Esse caso é exemplificado na linha que está comentada em relação ao fator **d2**.

ASSIGN

```

init(d10.state) := m;
init(d1.state)  := m;
init(d2.state)  := m;
--init(d2.state) := {l,m,h};
init(d12.state) := m;
init(d4.state)  := l;
init(d5.state)  := m;
init(d6.state)  := m;
init(d9.state)  := l;
init(d13.state) := m;
init(d20.state) := l;
init(g.state)   := l;

```

As assertivas lógicas a serem verificadas pelo verificador de modelos são definidas na seção **SPEC**. Neste caso, para determinar se o risco global do projeto pode chegar ao nível alto a assertiva em lógica TCL a ser verificada pelo analisador de modelos fica expressa em linguagem natural: não existe algum estado em que o nível do risco global (representado pela variável **g.state**) é alto. Se existir algum estado que por consequência da sucessão da ocorrência de eventos e influência dos fatores de risco o risco global do projeto (variável **g.state**) for alto, então os valores das variáveis ao longo da computação dos estados são apresentados como prova de que a propriedade é falsa. Caso a propriedade for verdadeira, o analisador de modelos simplesmente confirma que propriedade é verdadeira. A garantia da avaliação completa de todos os estados possíveis é expressa pela diretiva **FAIRNESS**. A avaliação de propriedades, como a avaliação de um fator de risco específico, está exemplificado como a avaliação do fator de risco **d2** na linha comentada.

SPEC

```

!EF(g.state = h)
--!EF(d2.state = h)

```

FAIRNESS

```

running

```

3 Avaliação

A avaliação do método proposto foi realizada através de entrevistas e experimentos com usuários. O objetivo foi avaliar o novo método para análise de riscos, identificando pontos positivos e negativos, possíveis melhorias e viabilidade de uso. Os participantes são gerentes de projeto ou envolvidos (*stakeholders*) no desenvolvimento de projetos de software.

A entrevista foi realizada através de ligação de voz na Internet e *chat*, utilizando o programa *Skype*®³, além de formulários eletrônicos disponibilizados pelo *Google Docs*⁴. A aceitação do termo de consentimento de participação no experimento, disponível no Apêndice A, foi realizada por uma mensagem de confirmação por e-mail. O áudio das conversas foi gravado para permitir análise mais detalhada da entrevista. As respostas dos questionários estão disponíveis no próprio sistema do *Google Docs*.

O tempo total estimado para o experimento foi de 55 minutos, embora nenhuma entrevista tenha sido interrompida por causa do tempo. O experimento mais demorado teve duração de 01h45min e na média eles levaram cerca de 01h30min, excesso que foi motivado principalmente pelo interesse dos entrevistados em conhecer detalhes sobre a metodologia.

Foram realizadas quatro entrevistas. Cada entrevista consistiu de três partes, mais um preâmbulo. O preâmbulo contém informações gerais sobre o entrevistado. A primeira parte é composta de três perguntas a respeito da disciplina de análise de riscos. O experimento é realizado na segunda parte, composto de uma breve descrição da metodologia e simulação de uso. Por fim, são realizadas mais três perguntas na terceira parte em que o entrevistado pode expressar opiniões a respeito do método experimentado. Os documentos utilizados na entrevista estão disponíveis como apêndice deste trabalho. No Apêndice B está o questionário utilizado na entrevista. As lâminas utilizadas na explicação do método e a *checklist* utilizada no experimento estão nos Apêndices C e D, respectivamente.

O perfil dos entrevistados é bastante variado, o que possibilitou ter uma diversidade de opiniões e enriquecendo a avaliação, de acordo com os objetivos do experimento. A seguir, são relatados resumos das entrevistas.

O primeiro entrevistado possui formação na área de humanas e coordena projetos de software para mineração de dados em redes sociais na Internet. Mesmo não tendo formação principal na área tecnológica, o entrevistado não teve dificuldades em entender o método proposto ou em realizar o experimento. A maior dificuldade encontrada foi a falta de definições precisas das metas dos projetos em que trabalha, pois utiliza-se um desenvolvimento baseado em métodos ágeis de desenvolvimento com alta volatilidade dos requisitos. Esta instabilidade nas definições do projeto e das tecnologias relacionadas, motivado principalmente por se tratar de projetos de uma empresa de pequeno porte (*start-up*), levou a avaliações de alto risco dos projetos, o que já era sabido. Para este entrevistado a metodologia não traz benefícios em curto prazo, pois ele precisa de uma melhor organização dos projetos de tal sorte a não ter nenhum risco alto associado aos fatores de risco chave. A criatividade deste entrevistado se destaca dos demais na resposta para a pergunta sobre o que ele gostaria de saber do analisador de modelos. A resposta foi: “estado atual, futuro, comparações com outras empresas do meu segmento”. A partir das informações que ele forneceu ao sistema, ele esperava que fosse possível saber o estado de risco atual, permitir simulações sobre o futuro e até comparações do estado de risco com projetos de concorren-

³<http://skype.com>

⁴<http://docs.google.com>

tes. Essas expectativas são genuínas e válidas, mas vão muito além do que é possível fazer por enquanto. Fica registrado como ideia para trabalhos futuros na metodologia. Sobre o questionário o entrevistado ainda disse que o questionário não é intuitivo e é difícil de entender, com diversas perguntas que não se adequam a sua realidade.

O segundo entrevistado é um desenvolvedor de sistemas sênior, com doutorado em tecnologia, que atua em projetos de grande porte, mas sem processo bem definido. A metodologia foi facilmente entendida e a *checklist* preenchida. Uma dificuldade compartilhada por todos os entrevistados foi começar a preencher o questionário, o que demonstra que há um esforço significativo para aprender como responder a *checklist*. Para contornar essa dificuldade foram apresentados exemplos de respostas. As respostas da *checklist* foram bem elaboradas, com a definição de diversas propriedades a serem avaliadas pelo analisador de modelos. A tradução das perguntas em assertivas lógicas foi realizada pelo entrevistador, mas futuramente é necessário que haja um sistema para auxiliar o usuário a criar expressões lógicas. O entrevistado gostou da metodologia proposta e até usaria em seus projetos, mas sente uma urgência maior em relação a um processo a ser seguido no desenvolvimento do projeto.

O terceiro entrevistado é um gerente de projetos de pesquisa que já atuou mais de ano como analista de sistemas sênior de uma grande empresa de software, e também possui doutorado em tecnologia. Sua experiência como analista de sistemas foi bastante rica em relação a processos e metodologias em projetos. Possui aprofundado conhecimento de metodologias ágeis, especialmente o *SCRUM*. Demonstrou-se muito interessado em aprender sobre a disciplina de análise de risco, a qual tinha apenas vaga noção a respeito, apesar de fazer intuitivamente análise de risco de seus projetos. A todo o momento perguntava se poderia utilizar a metodologia proposta para realizar a avaliação de seus projetos de pesquisa. O experimento foi rico na ilustração de situações em que o projeto avaliado alcança risco alto, corroborando exatamente o maior medo do entrevistado (ocorrência de um determinado evento). Por se tratar de um experimento feito rapidamente, foi recorrente em todos os experimentos que os resultados obtidos pelo analisador de modelos foram triviais, sem ocorrências de muitos eventos para que o risco geral do projeto fosse alto. Mesmo assim, o entrevistado se demonstrou confiante na metodologia desenvolvida. Ele também solicitou que se criassem perguntas direcionadas a projetos de pesquisa, pois as perguntas do questionário não estavam aderentes a sua realidade. O entrevistado também encontrou muita dificuldade em entender as perguntas da *checklist*. Além disso, também citou a escala de apenas três valores, pois na dúvida ele marcou a resposta com risco médio, a qual ele chamou de “resposta em cima do muro” (também observado pelo segundo entrevistado). Apesar das críticas, a avaliação geral da metodologia foi bem positiva e o entrevistado ainda ressaltou que gostaria que o questionário fosse adaptado a sua realidade para ser utilizado de fato em seus projetos.

O quarto entrevistado é professor pós-graduado em engenharia de software e que eventualmente desenvolve projetos para o mercado, desempenhando o papel de gerente de projetos. Como possui formação na área, já estava familiarizado com conceitos de risco e análise de risco, apesar de após a explicação da proposta deste trabalho reconhecer que seu conhecimento estava desatualizado. A definição de análise de risco fornecida inicialmente na entrevista foi em relação a processos de desenvolvimento de softwares mais antigos, rigorosos e longe da sua realidade. Por esse motivo, o entrevistado também disse que não pratica análise de risco em seus projetos, pois prefere utilizar sua intuição. O experimento

com o método proposto permitiu revelar a dificuldade que gerentes de software poderão enfrentar ao identificar eventos significativos que representem o risco em seus projetos. No final do teste, o entrevistado gostou de ter participado do experimento por ter atualizado seus conhecimentos em relação à disciplina de análise de risco e ter conseguido usar a ferramenta, apesar de ainda preferir utilizar sua intuição no gerenciamento de seus projetos, que são de pequeno porte. Para projetos de maior porte, ele disse que usaria esta nova metodologia sim, pois considera útil também para documentar os riscos do projeto e compartilhar este conhecimento com a equipe.

A maioria dos entrevistados ressaltou a necessidade de adaptar a *checklist* à realidade de cada projeto. Isto já era esperado, pois o método proposto é um *framework* e cada modelo representa a realidade de um projeto. O experimento foi realizado com um mesmo modelo por causa da complexidade de se adaptar o modelo apresentado como exemplo. Embora, o experimento tenha sido comprometido pela falta de aderência ao projeto que estava sendo avaliado ficou claro que em um experimento de curta duração não é possível realizar uma avaliação se tiver que também criar o modelo. Contudo, considerando o fato de os entrevistados usarem projetos com que trabalha diariamente, a identificação dos fatores de risco usados na análise de risco foi facilitada com o modelo utilizado na avaliação e também foi possível perceber com maior realidade as necessidades dos entrevistados.

4 Conclusão

Este trabalho propõe um novo método para realizar análise de risco no desenvolvimento de projetos de software. Foi apresentado um modelo para reflexão computacional sobre análise de risco, que é baseado em uma máquina de estados finitos e permite a validação de propriedades definidas por lógica temporal TCL. A epistemologia utilizada na construção do método foi detalhada, bem como foi mostrada a implementação de um método para análise de risco na linguagem de especificação formal utilizada pelo verificador de modelos SMV.

Foram realizadas entrevistas e experimentos com gerentes de projeto ou envolvidos (*stakeholders*) no desenvolvimento de projetos. O objetivo foi avaliar o método proposto para análise de riscos, identificando pontos positivos e negativos, possíveis melhorias e viabilidade de uso. Os entrevistados avaliaram positivamente o método proposto e ressaltaram a necessidade de adaptar a *checklist* (modelo) à realidade de cada projeto.

Um dos entrevistados disse que usaria a metodologia proposta neste trabalho se já houvesse catalogado fatores de risco chave para projetos de pesquisa. Portanto, este é um desdobramento factível desta pesquisa. Outra melhoria que pode ser agregada ao trabalho é considerar múltiplos *stakeholders* respondendo o questionário para um mesmo projeto com unificação automatizada de eventos.

Referências

Alberts, C. J. & Dorofee, A. J. (2009), A framework for categorizing key drivers of risk, Technical Report CMU/SEI-2009-TR-007, Software Engineering Institute, Pittsburgh, PA.

- Bakker, K., Boonstra, A. & Wortmann, H. (2009), ‘Does risk management contribute to it project success? a meta-analysis of empirical evidence’, *International Journal of Project Management*.
- Bannerman, P. L. (2008), ‘Risk and risk management in software projects: A reassessment’, *J. Syst. Softw.* **81**(12), 2118–2133.
- Clarke, E. M., Emerson, E. A. & Sistla, A. P. (1986), ‘Automatic verification of finite-state concurrent systems using temporal logic specifications’, *ACM Trans. Program. Lang. Syst.* **8**(2), 244–263.
- Committee, P. S. (2008), *A guide to the project management body of knowledge (PMBOK Guide) (4th ed.)*, Project Management Institute (PMI), Newtown Square, PA.
- Emerson, E. A. (1990), Temporal and modal logic, in J. van Leeuwen, ed., ‘Handbook of Theoretical Computer Science’, Elsevier, pp. 996–1072.
- Keil, M., Cule, P. E., Lyytinen, K. & Schmidt, R. C. (1998), ‘A framework for identifying software project risks’, *Commun. ACM* **41**(11), 76–83.
- McMillan, K. L. (1992), ‘The SMV system’.
- Tiwana, A. & Keil, M. (2004), ‘The one-minute risk assessment tool’, *Commun. ACM* **47**(11), 73–77.
- Varnell-Sarjeant, J. F. (2008), ‘Managing a man-rated software development program via risk mitigation’, *SIGSOFT Softw. Eng. Notes* **33**(4), 1–8.
- Wallace, L., Keil, M. & Rai, A. (2004), ‘Understanding software project risk: a cluster analysis’, *Inf. Manage.* **42**(1), 115–125.
- Warkentin, M., Moore, R. S., Bekkering, E. & Johnston, A. C. (2009), ‘Analysis of systems development project risks: an integrative framework’, *SIGMIS Database* **40**(2), 8–27.

A Termo de Consentimento



Departamento de Informática, PUC-Rio
Rua Marquês de São Vicente, 225
Gávea – Rio de Janeiro – RJ – 22451-900
Tel. (21) 3114-1500 r. 3323

Termo de Consentimento

Reflexão computacional é a habilidade do sistema computacional raciocinar sobre seu comportamento, podendo ou não modificá-lo. Tal reflexão é extremamente útil para entender e solucionar problemas em sistemas computacionais. Esta pesquisa faz parte de um estudo multidisciplinar sobre linguagens de representação para reflexão computacional, particularmente, estamos interessados em avaliar uma nova metodologia para análise de riscos. O objetivo desse experimento é identificar pontos positivos e negativos, possíveis melhorias e viabilidade de uso do método proposto.

Por isto, **convidamos você a colaborar com nossa pesquisa**, composta de três etapas, mais preâmbulo:

1. Preâmbulo.
2. Introdução.
3. Experimento.
4. Conclusão.

Para decidir sobre sua participação, é importante que você tenha algumas informações adicionais:

1. Os dados coletados serão acessados somente pela equipe desta pesquisa. A entrevista será gravada, apenas para que possamos analisar com cuidado os dados coletados.
2. A divulgação dos resultados de nossa pesquisa – exclusivamente para fins acadêmicos – pauta-se no respeito à privacidade, e **o anonimato dos participantes é preservado em quaisquer documentos que elaborarmos.**
3. O consentimento para participação é uma escolha livre, e esta participação pode ser interrompida a qualquer momento, caso você precise ou deseje.

De posse das informações acima, você:

Consinto em participar.

Participante: _____

Assinatura: _____

Rio de Janeiro, 07 de dezembro de 2009.

B Formulário da entrevista

Experimento sobre Nova Metodologia para Análise de Riscos em Projetos

Objetivo: Avaliar nova metodologia para análise de riscos, identificando pontos positivos e negativos, possíveis melhorias e viabilidade.

Perfil dos participantes: Gerente de projeto, e stakeholders em projetos de software.

Resumo: Entrevista consiste de 3 partes, mais um preâmbulo. A primeira parte é composta de três perguntas gerais sobre a disciplina de análise de riscos. O experimento é realizado na segunda parte, composto de uma breve descrição da metodologia e simulação de uso. Por fim, são realizadas mais três perguntas na terceira parte em que o entrevistado pode expressar opiniões a respeito da ferramenta. O preâmbulo contém informações sobre o entrevistado.

Tempo total estimado: 55 minutos.

[Continue »](#)

Powered by [Google Docs](#)

[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)

Experimento sobre Nova Metodologia para Análise de Riscos em Projetos

* Required

Preâmbulo

O objetivo deste questionário é saber mais sobre a formação e experiência geral do participante.
Tempo estimado: 6 minutos.

Nome

(opcional)

Escolaridade *

Profissão *

Certificações *

Indústria da empresa em que trabalha *

Experiência com gerência de projetos (anos, meses) *

Outras informações que julgar relevante

« Back

Continue »

Powered by [Google Docs](#)

[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)

Experimento sobre Nova Metodologia para Análise de Riscos em Projetos

* Required

Introdução

O objetivo deste questionário é saber mais sobre o nível de conhecimento e experiência do participante em relação a disciplina de análise de risco. Tempo estimado: 9 minutos.

Você sabe o que é análise de riscos em projetos? Explique. *

Você realiza análise de riscos nos seus projetos? Se sim, explique. *

Você gostaria de alguma metodologia que lhe ajudasse a realizar análise de risco em projetos? *

« Back

Continue »

Powered by [Google Docs](#)

[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)

Experimento sobre Nova Metodologia para Análise de Riscos em Projetos

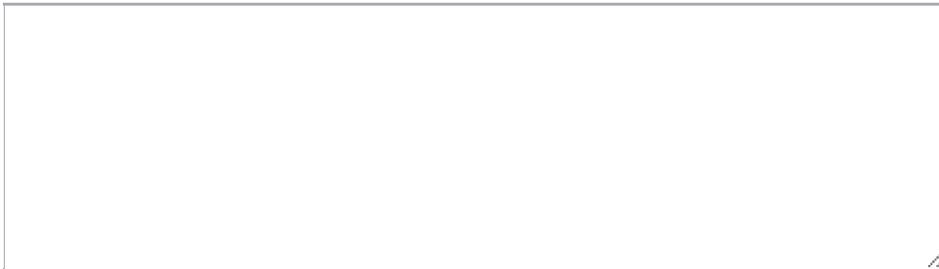
Experimento

Realizar simulação de análise de risco do projeto que está trabalhando no momento, utilizando a nova metodologia proposta. Tempo estimado: 30 minutos.

Apresentar nova metodologia para análise de risco

Descreva brevemente sobre o projeto em questão.

Se não for possível por questões de confidencialidade industrial, relate um caso hipotético similar ao que será realizado a análise.



Responder o questionário de análise de risco.

Responder no questionário apropriado.

Powered by [Google Docs](#)

[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)

Experimento sobre Nova Metodologia para Análise de Riscos em Projetos

* Required

Conclusão

Tempo estimado: 10 minutos.

O que você achou da nova metodologia? Pontos positivos e negativos. *

Você considera que esta ferramenta pode melhorar o seu trabalho? *

Outros comentário.

« Back

Submit

Powered by [Google Docs](#)

[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)

C Lâminas utilizadas para explicar a nova metodologia

Risk Analysis in Projects: an approach for computational reflection based on logic

Darlinton Carvalho
darlinton@gmail.com

Project

Creation of a tool to help users in the risk analysis of software development project, using a novel representation for computational reflection.

- Domain
 - Software development
- Task
 - Risk Analysis
- Development
 - Tool

Risk Analysis

- What is risk?
- What is risk management?
- Improving software projects
- Does risk management contribute to IT project success?

 Bannerman, P. L. 2008. Risk and risk management in software projects: A reassessment. *J. Syst. Softw.* 81, 12 (Dec. 2008), 2118-2133. DOI= <http://dx.doi.org/10.1016/j.jss.2008.03.059>

 De Bakker, K et al. Does risk management contribute to IT project success? A meta-analysis of empirical evidence. *Int J Project Manage* (2009), doi:10.1016/j.ijproman.2009.07.002

Risk Analysis

- What is risk?

$$R = P \times I$$

- **R** is risk exposure attributable to a particular risk factor
- **P** is the probability the event will be realized
- **I** is the impact or magnitude of the event



- Classical decision theory X a modern view
 - 80% of managers consider only negative outcomes as 'risk' (March and Shapira (1987))
 - In Software Projects **I** is only negative

Risk Analysis

- What is risk management?
 - Set of principles and practices aimed at identifying, analyzing and handling risk factors
 - To improve the chances of achieving a successful project outcome and/or avoid project failure (Boehm 1989, 1991; Charette, 1989; Kerzner, 2003)
 - Checklists, analytical frameworks, process models, risk response strategies.



Risk Analysis

- Improving software projects
 - Literature reports that risk management can significantly improve software projects outcomes.
 - It also reports that risk management is not always well-applied in practice.
 - Study of project management maturity based on PMI's knowledge areas found that the risk knowledge area had the lowest maturity of all knowledge areas in the IS industry, and the IS industry was the lowest of the four in the study.
 - 75% of project managers did not follow any detailed risk management approach, and only vaguely understood the software risk concept and its managerial implications.



Risk Analysis

- Does risk management contribute to IT project success?

Fig. 4. Traditional view of how project success is measured.

10/12/2009 Darlinton Carvalho @ LES/PUC-Rio 7

Novel approach

A Framework for Logic Analysis of Key Drivers of Risk

- Based on the work:

Alberts, C.J. & Dorofee, A.J. A Framework for Categorizing Key Drivers of Risk. TECHNICAL REPORT, ESC-TR-2009-007, CMU/SEI-2009-TR-007, 2009

10/12/2009 Darlinton Carvalho @ LES/PUC-Rio 8

Risk Analysis Theory

Figure 1. Components of Risk

Figure 2. A Detailed View of Risk

Figure 3. A Systemic View of Risk

10/12/2009 Darlinton Carvalho @ LES/PUC-Rio 9

Risk Analysis Theory

Figure 4. Relationship Among Key Objectives, Causes, Conditions, and Future Events

Figure 5. Six Categories of the Linear Framework

10/12/2009 Darlinton Carvalho @ LES/PUC-Rio 10

Risk Analysis Theory

Figure 6. Relationship Among the Driver Categories

Alberts, C.J. & Dorofee, A.J. A Framework for Categorizing Key Drivers of Risk. TECHNICAL REPORT, ESC-TR-2009-007, CMU/SEI-2009-TR-007, 2009

10/12/2009 Darlinton Carvalho @ LES/PUC-Rio 11

Risk Analysis Theory

Figure 7. Relationship Between Key Objectives and Drivers

10/12/2009 Darlinton Carvalho @ LES/PUC-Rio 12

Risk Analysis Theory

LES

Table 1: (Revised) Set of Drivers for Software Programs

I. Process	The process being used to develop and deploy the system is sufficient.	The process being used to develop and deploy the system is insufficient.	Preparation
------------	--	--	-------------

Table 2: Driver Question and Range of Responses

Driver Question	Answer				
	No	Likely no	Equally likely	Likely yes	Yes
I. Is the process being used to develop and deploy the system sufficient? Consider: process design; measurements and controls; process efficiency and effectiveness; acquisition and development life cycle; teaming; Fit.	☐	☐	☐	☐	☐

10/12/2009 Darlinton Carvalho @ LES/PUC-Rio 13

Risk Analysis Theory

LES

Table 4: Driver Value Ratings

Answer	Definition	Velocity	
		Probability of Success State	Probability of Failure State
Yes	The answer is simply "yes" with no additional nuance. This is the state of no probability for the answer "not to take". → 0% Probability of yes	Minimum	Maximum
Likely yes	The answer is "likely yes". There is some chance for the answer "not to take". → 25% Probability of yes	High	Low
Equally likely	The answer is "just a little bit yes" or "no". → 50% Probability of yes	Medium	Medium
Likely no	The answer is "likely no". There is some chance for the answer "not to take". → 75% Probability of yes	Low	High
No	The answer is simply "no" with no additional nuance. This is the state of no probability for the answer "not to take". → 100% Probability of yes	Maximum	Minimum

10/12/2009 Darlinton Carvalho @ LES/PUC-Rio 14

Risk Analysis Theory

LES

I. Is the process being used to develop and deploy the system sufficient? Consider: process design; measurements and controls; process efficiency and effectiveness; acquisition and development life cycle; teaming; Fit.	☐	☐	☐	☐	☐
---	---	---	---	---	---

Notes:

- Mostall projects have a 80% history of delivering on-time.
- The process for integration testing is fairly immature. Historically, integration testing has used "walkthrough" agreements between a few managers who already know each other. With this system, teams are managed and team leads who have never worked together and there are other barriers in place that make "walkthrough" agreements difficult.
- There are a lot of local test engineers (WET).
- The program required a significant change in our architecture. There will be new training needed for the new process.
- QA did not have a chance to review the new and revised processes before they went out into production.
- The program was developed for requirements just last week.

Figure 10: Revised Table

10/12/2009 Darlinton Carvalho @ LES/PUC-Rio 15

Risk Analysis Theory

LES

10/12/2009 Darlinton Carvalho @ LES/PUC-Rio 16

Risk Analysis Theory

LES

Figure 11: Driver Profile¹

10/12/2009 Darlinton Carvalho @ LES/PUC-Rio 17

Risk Analysis Theory

LES

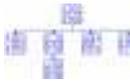
Figure 12: Example Risk Profile

10/12/2009 Darlinton Carvalho @ LES/PUC-Rio 18

Extension



- Multiple users answer the questionnaire
- Answer rationale is defined by events
 - changes over the answers
- Analysis is made over the composition of users' answers.
 - ?
- Creation of a logic model to represent the questionnaire answers
 - Allow logic queries.
- Risk analysis is done by Model Checker assertive.
 - Is there any way to have Answer No (high risk)?
 - The answer is true if there isn't a case.
 - The answer is false and an instance is given as result. The result explains a scenario where the risk can be high.



10/12/2009
Darlington Carvalho @ LES/PUC-Rio
19

Extension



Table 2 - Driver Question and Range of Response

Driver Question	Answer				
	No	Probably	Equally likely	Probably yes	Yes
2. Is the process being used to develop and deploy the system sufficient? <small>Consider: process design (requirements and controls, process efficiency and effectiveness), acquisition and development life cycles, training.</small>	<input type="checkbox"/>				

Rationale

- event1
- event2
- event3

Rationale

- event4
- event5
- event6

10/12/2009
Darlington Carvalho @ LES/PUC-Rio
20

Backup



10/12/2009
Darlington Carvalho @ LES/PUC-Rio
21

Model Checking: O que é ?



- Especificar
 - O sistema como um sistema de transição (finito em geral).
 - Uma propriedade como uma sentença em uma linguagem de consulta.
- Estabelecer que o sistema é um modelo da sentença

sistema |= *sentença*

Obs: Tarefa realizada por varredura no espaço de busca

- Usualmente não há necessidade de teorias auxiliares
- Normalmente, as únicas suposições dizem respeito às propriedades dinâmicas do sistema (branching vs. linear time, fairness, liveness)
- Funciona melhor com sistemas voltados ao controle e com um pequeno espaço de estados

TECMF
Profs Clarisse, Renato e Hermann
Métodos Formais em Reflex. Comput.

Risk Analysis



Risk and Managers

- Managers:
 - concern with the magnitude of the potential loss than the probability it will occur.
 - **I is more important than P**
 - Prefer verbal characterizations of risk than probabilistic representation because they are skeptical that the broad dimensionality of risk can be reduced to a single number
 - **Managers don't like P**
 - Tend not to accept risk estimates given to them because they see risk as subject to control.
 - **They manage R**



10/12/2009
Darlington Carvalho @ LES/PUC-Rio
23

Risk Analysis – new approaches



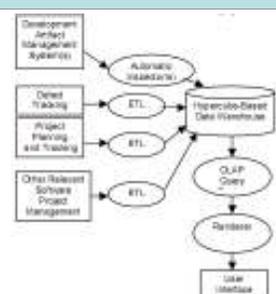


Figure 6: Typical data collection architecture for software project decision support.

Masticola, S. P. 2007. Lightweight Risk Mitigation for Software Development Projects Using Repository Mining. In Proceedings of the Fourth International Workshop on Mining Software Repositories (May 20 - 26, 2007). International Conference on Software Engineering, IEEE Computer Society, Washington, DC, 13. DOI= <http://dx.doi.org/10.1109/MSR.2007.16>

10/12/2009
Darlington Carvalho @ LES/PUC-Rio
24

D *Checklist* de fatores de risco chave utilizada no experimento



Darlinton Carvalho <darlinton@gmail.com>

Key drivers risk evaluation

1 message

darlinton@gmail.com <darlinton@gmail.com>

Wed, Dec 9, 2009 at 9:21 AM

To: darlinton@gmail.com

If you have trouble viewing or submitting this form, you can fill it out online:

<https://spreadsheets.google.com/viewform?formkey=dFp1SF9RZWItbUlfafNTcmdXZ082U0E6MA>

Key drivers risk evaluation

All responses will be anonymous or you can type your name here. Thank you for your time!

Name and role

Example: John (manager), Joe (developer)

1. Program Objectives (Objectives) *

Program objectives (product, cost, schedule) are realistic and achievable.

rationale 1

describe the event which must happen to change the answer - mask: actual state; next state; description

2. Plan (Preparation) *

The plan for developing and deploying the system is sufficient.

rationale 2

describe the event which must happen to change the answer - mask: actual state; next state; description

4. Task Execution (Execution) *

Tasks and activities are performed effectively and efficiently.

rationale 4

describe the event which must happen to change the answer - mask: actual state; next state; description

5. Coordination (Execution) *

Activities within each team and across teams are coordinated appropriately.

rationale 5

describe the event which must happen to change the answer - mask: actual state; next state; description

6. External Interfaces (Execution) *

Work products from suppliers, partners, or collaborators will meet the program's quality and timeliness requirements.

rationale 6

describe the event which must happen to change the answer - mask: actual state; next state; description

9. Facilities and Equipment (Execution) *

Facilities and equipment are sufficient to support the program.

rationale 9

describe the event which must happen to change the answer - mask: actual state; next state; description

10. Organization Conditions (Environment) *

Enterprise, organizational, and political conditions are facilitating completion of program activities.

rationale 10

describe the event which must happen to change the answer - mask: actual state; next state; description

12. Event Management (Resilience) *

The program has sufficient capacity and capability to identify and manage potential events and changing circumstances.

rationale 12

describe the event which must happen to change the answer - mask: actual state; next state; description

13. Requirements (Result) *

System requirements are well understood.

rationale 13

describe the event which must happen to change the answer - mask: actual state; next state; description

20. Certification and Accreditation (Result) *

The system will be appropriately certified and accredited for operational use.

rationale 20

describe the event which must happen to change the answer - mask: actual state; next state; description

questions

logic questions that you would like to do about this questionnaire. e.g.: risk states possible states

Submit

Powered by [Google Docs](#)

[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)
