Um Estudo sobre a Atividade de Elicitação de Requisitos em Projetos de Software da Área Espacial

Carlos Lahoz[1, 2] e João Batista Camargo Jr.[2]
[1] *Instituto de Aeronáutica e Espaço, Praça Mal. Eduardo Gomes, 50, 12228-904 São José dos Campos, Brasil*
lahoz@iae.cta.br
[2] *Escola Politécnica da Universidade de São Paulo, Av. Prof. Luciano Gualberto, 158 05508-900 São Paulo, Brasil*
joao.camargo@poli.usp.br

Resumo

Dentro do enfoque de que o software começa a atender cada vez mais um número crescente de funcionalidades de sistemas, é fundamental que a atividade de elicitação de requisitos assuma um papel decisivo no esforço de se alcançar um resultado satisfatório e seguro no projeto de um sistema. No que diz respeito a sistemas aeroespaciais críticos, onde a ambigüidade, a não completeza e a falta de requisitos podem provocar acidentes graves, envolvendo prejuízos econômicos, materiais e humanos, obriga a um tratamento mais cuidadoso sobre este assunto. Este artigo apresenta os resultados preliminares de um estudo sobre quais são os principais problemas que a atividade de elicitação de requisitos enfrenta atualmente em projetos espaciais no Brasil, segundo a visão dos autores. O foco principal do trabalho está na identificação dos requisitos de segurança em projetos de desenvolvimento de software.

1. Introdução

Segundo Kotonya e Sommerville [8] o termo engenharia de requisitos foi criado para cobrir todas as atividades envolvidas na descoberta, documentação, e manutenção de um conjunto de requisitos para um sistema baseado em computador. Para Laplante [10], a disciplina de engenharia de requisitos de software compete à determinação dos objetivos, funções, e restrições de um sistema de software e a representação dos seus aspectos de forma acessível para modelagem e análise. Obviamente que esta disciplina tem também como objetivo identificar requisitos que sejam corretos, completos e entendíveis para clientes e desenvolvedores. Pode-se dizer então, que o termo engenharia de requisitos, abrange tanto os assuntos relativos à análise e especificação de sistemas de informação (negócios) como o de análise e especificação de sistemas de engenharia. Sistemas espaciais são classificados em sistemas de engenharia, onde os requisitos envolvem hardware, software, procedimentos operacionais e processos: os chamados sistemas embarcados e sistemas de comando e controle [10].

O processo de engenharia de requisitos, ou seu nível de detalhamento, pode variar de uma organização para outra. Uma organização de desenvolvimento de projetos de software da área espacial pode não usar os mesmos processos de uma organização de tecnologia da informação. Mas de modo geral o processo de engenharia de requisitos, em qualquer tipo de sistema, abrange as atividades de elicitação, análise, documentação, validação e gerenciamento de requisitos.

O foco desta pesquisa está na atividade de elicitação de requisitos, que conforme proposto por Kotonya e Sommerville [8] e Sawyer e Kotonya [19], é o nome usual dado às atividades envolvidas na descoberta dos requisitos de um sistema. Para a realização desta atividade são utilizadas técnicas como, por exemplo, entrevistas, observação, análise de cenários, prototipação. Seu principal objetivo é extrair informações sobre o problema a ser resolvido, o serviço que o sistema proposto deve disponibilizar, os requisitos de desempenho, as restrições de hardware, entre outros.

Quando falamos de elicitação de requisitos e, especificamente de requisitos de segurança, primeiramente pensamos sobre requisitos funcionais que apresentam ramificações de segurança crítica [2]. Enquanto normalmente os requisitos especificam o que o sistema deve fazer ou espera-se que faça, os requisitos de segurança especificam o que o sistema não deve fazer, ou deve se prevenir para que aconteça [16]. Em sistemas onde a questão da segurança é crítica, requisitos funcionais são tanto aqueles que podem causar acidentes se não forem implementados ou se o forem incorretamente. Já os requisitos não funcionais são aqueles que não são especificamente requeridos

como funcionalidades do sistema. Eles apresentam restrições ao produto a ser construído e ao processo de seu desenvolvimento, e suas especificações de restrição externa ao qual o produto deve atender [10].

Este trabalho introduz primeiramente alguns estudos e relatórios sobre problemas envolvendo software e requisitos em organizações da área espacial, como o do acidente com o foguete Ariane 5, da agência espacial francesa e o estudo do acidente que ocorreu com a sonda Mars Climate Orbiter (MCO), da agência espacial americana. Em seguida é apresentado, sob o ponto de vista dos autores, quais são os principais obstáculos que as atividades da engenharia de requisitos enfrentam nesta área, especificamente na atividade de elicitação. Depois, são feitos comentários sobre os envolvidos no processo de elicitação de requisitos dentro deste contexto. Também é apresentada uma primeira forma de abordar o problema de segurança através de uma classificação dos requisitos. Estudos futuros sobre possíveis formas de abordar a atividades de elicitação de requisitos e segurança são apresentados em seguida. Finalmente, algumas considerações sobre como superar obstáculos, segundo a estratégia dos autores, é mostrado.

2. Requisitos e Projetos Espaciais

Existe atualmente, na literatura técnica especializada, um farto material tratando sobre as disciplinas da engenharia de software e seus problemas de projeto. Na área espacial trabalhos recentes se destacam sobre este tipo de estudo: as investigações da professora do Departamento de Aeronáutica e Astronáutica e de Engenharia de Sistemas do Massachusetts Institute of Technology (MIT), Nancy Leveson, bem como o relato do pesquisador Kjeld Hjortnaes, do European Space and Technology Centre (ESTEC), e chefe do On-board Software Systems Section (TOS-SEM). Tanto nas investigações de acidentes espaciais envolvendo software, bem como a análise de Hjortnaes de mais de 18 revisões técnicas dos projetos de desenvolvimento de software da European Space Agency (ESA), foram levantadas questões de problemas envolvendo requisitos, engenharia de sistemas e de software.

De um modo geral, foram observados nestes estudos, dentre outros problemas, práticas inadequadas de especificações de requisitos ou um conhecimento insuficiente do sistema, levando a um entendimento pobre sobre seu contexto e seus requisitos operacionais.

No relatório da investigação do acidente do foguete Ariane 501 [12], estudado por Leveson, foi apontado práticas pobres de especificação de requisitos. No da investigação do acidente da nave Mars Climate Orbiter (MCO) e da Mars Polar Lander (MPL) [14] [15] recomendou-se treinamento mais adequado das equipes de desenvolvimento devido a problemas na especificação dos requisitos do sistema. Hjortnaes observa, na sua análise dos projetos da ESA, uma falta de maturidade e estabilidade da "baseline" de requisitos de software quando o processo de desenvolvimento se inicia. Ele cita ainda que, em muitos dos relatórios analisados, o desenvolvimento dos requisitos não foi conduzido de forma adequada ou não existia um entendimento correto dos mesmos. Ambos autores são unânimes em dizer que a Engenharia de Software deve ter uma participação mais efetiva no time de Engenharia de Sistemas.

Um ponto importante que deve ser salientado em projetos desta natureza é a questão da segurança. No caso específico do acidente do terceiro protótipo do Veículo Lançador de Satélites (VLS-1 V03) [20], um estudo independente proposto por Almeida e Johnson [1] reforça que no relatório oficial da investigação é citado a pouca atenção ao desenvolvimento de uma "cultura de segurança" apropriada ao projeto. Faltaram revisões, auditorias externas e validação por especialistas de avaliação de risco durante o desenvolvimento do projeto. Almeida e Johnson argumentam ainda que em algumas áreas do programa espacial brasileiro não há o reconhecimento da importância do gerenciamento de segurança de sistemas, como é recomendado nas indústrias da área de espaço da América do Norte e Europa.

Levando em consideração estes estudos, observou-se que em diversos projetos, mesmo de agências espaciais com um alto nível de maturidade em termos de profissionais e infra-estrutura, existem ainda diversos problemas de ordem operacional (processos), técnica (tecnologia) e humana (pessoas) [9]. Seguindo esta abordagem, sob o ponto de vista de tecnologia, processos e pessoas, é possível mapear as principais barreiras enfrentadas no levantamento de requisitos em projetos espaciais, e em seguida buscar os caminhos para superar estes obstáculos.

3. Tecnologia, Processos e Pessoas

Na visão de tecnologia é observado o uso inadequado, ou mesmo inexistente de uma técnica de elicitação e de representação de requisitos. Técnicas como entrevista, prototipação ou mesmo o estudo de documentação, muitas vezes são mal empregadas, apresentando resultados não muito confiáveis.

Também, o uso inapropriado de modelos de representação dos componentes do sistema e suas interações podem não destacar adequadamente as dependências comportamentais advindas da especificação destes componentes, gerando problemas que podem ser descobertos (quando descobertos) somente em fases muito avançadas do desenvolvimento de um sistema. Caso típico de perdas financeiras, de

atraso no cronograma de projeto e perda total do crédito da equipe.

A falta de uma abordagem sistêmica de segurança em sistemas de software, principalmente em ambientes de pouca maturidade em cultura de segurança, pode não ser percebido em um primeiro momento nas atividades de elicitação, mas fatalmente afetará os estágios mais avançados do desenvolvimento, ou, pior ainda, somente quando o sistema estiver em operação.

A falta ou o uso inapropriado de ferramentas de engenharia de software e de engenharia de requisitos, como disponível em ferramentas CASE, nos verificadores de modelos e em outros recursos de apoio à identificação, representação e consistência de requisitos, geram erros e mal entendidos em projeto que conduzem a ignorância, desconfiança, constrangimento e até descrédito da equipe e do projeto.

Na visão de processo, a indefinição ou pobreza de modelos de processo, assim como uma abordagem precária de segurança, está fortemente relacionada a pouca maturidade da equipe e da organização ao qual pertence. Muitas vezes, por razões externas ao projeto, como pessoal sem a qualificação necessária, estrutura organizacional ineficiente ou mesmo pressões de tempo e de recursos, as atividades do processo de elicitação de requisitos são menosprezadas em detrimento de outras atividades, como por exemplo, as relacionadas diretamente ao desenvolvimento de código.

Mesmo com um processo de elicitação estabelecido, sem um monitoramento ou seu uso de forma parcial, não possibilitará o acompanhamento das alterações surgidas durante a vida de um requisito desde sua fase de elicitação e ao longo do ciclo de vida do desenvolvimento do sistema.

Outro obstáculo presente em projetos, não só da área espacial, é a comunicação inadequada entre os envolvidos. Sem o auxilio de algum tipo de mecanismo ou de uma ferramenta automatizada, o processo de comunicação será ineficiente, não permitindo que durante a fase de elicitação, os requisitos extraídos pelos envolvidos sejam avaliados de forma abrangente, de maneira a permanecerem estáveis durante a vida do projeto.

Na visão de pessoas, diferentes enfoques de ordem cultural entre os participantes, sejam engenheiros de sistema como de software, podem criar um obstáculo quase intransponível. Software é uma disciplina relativamente nova, segundo a visão dos engenheiros de sistema, e muitas vezes, a participação da equipe de software nas atividades de elicitação dos requisitos de sistema, não são consideradas tão importantes.

Outro ponto crítico é o entendimento limitado do domínio do problema, que aliado a uma visão parcial dos envolvidos, prejudica qualquer técnica de elicitação. Sem o entendimento claro e preciso do domínio do problema, é difícil interagir com os participantes e ser capaz de ir além da visão particular de como cada um enxerga o sistema.

Ainda mais grave é um gerenciamento de projeto que não considera a importância das atividades de elicitação. A gerência de um projeto tem que equilibrar os interesses dos participantes no processo de elicitação, definir as prioridades dos usuários e promover o consenso entre os envolvidos. Sem este equilíbrio, forças antagônicas de interesse aparentemente comuns, podem gerar, como uma conseqüência extrema, a morte lenta e gradual de um projeto.

4. Envolvidos no Processo

Segundo Gonzáles [5] existem basicamente duas comunidades que estabeleceram métodos para capturar, especificar e gerenciar requisitos: a engenharia de sistemas e a engenharia de software. Apesar de abordagens técnicas e culturais diferentes, estas comunidades possuem muitos elementos em comum e podem aprender uma com a outra. Também não se pode excluir do processo de elicitação de requisitos de sistemas espaciais outros envolvidos que representam papéis importantes: o próprio cliente, o engenheiro de segurança, o gerente de projeto e o usuário do sistema.

Cliente: é quem define o sistema no seu nível mais abrangente. Geralmente em projetos espaciais são representados pelas agências governamentais ou pelos dirigentes de alto nível de um programa desta natureza. O papel desempenhado pelo cliente é de observar o sistema dentro de um contexto estratégico-político diferente do observado pelos outros envolvidos e pode não ter muita influencia nos rumos tomados por um projeto no seu nível mais prático de desenvolvimento e acompanhamento.

Engenheiro de Sistema: representa o papel de quem promove o consenso entre as pessoas envolvidas no sistema e desenvolve um entendimento mais global da solução do problema. É avesso aos riscos de projeto. Usa linguagem natural para prospectar soluções e procura balancear os métodos de engenharia e gerenciamento para alcançar seus objetivos. Entende a disciplina de requisitos como necessária, mas carece de uma visão metodológica da disciplina.

Engenheiro de Requisitos: neste caso específico, ele é oriundo do time de engenharia de software e é quem modela o problema através de um conjunto de formalismos e geralmente imagina o software como todo o sistema. Faz especificação e aplica métodos que

produzirão resultado independente das pessoas envolvidas. Apesar de possuir uma abordagem mais metódica da disciplina de requisitos, ainda falta uma visão mais abrangente do contexto do sistema e do software como um todo. Pode, muitas vezes, nem existir seu papel em uma organização, sendo assumido por algum outro tipo de engenheiro.

Engenheiro de Software: toda a equipe de projeto de software envolvida com os requisitos, como modeladores, programadores, testadores, equipe da qualidade, etc. Possuem papel importante na manipulação dos requisitos, podendo em muitos casos, devido a uma interpretação errônea ou parcial de sua funcionalidade, comprometer todo o projeto.

Engenheiro de Segurança: o papel do engenheiro de segurança, tanto na engenharia de sistemas como na de software, ou não existe ou é recurso raro no mercado. Este profissional geralmente tem (alguma) experiência com implementação de políticas de segurança, com adoção e documentação de padrões, monitoração dos procedimentos de segurança, análise de risco, de segurança e investigação de acidentes. Pode não ter um bom conhecimento do hardware e do software utilizado em um projeto, e nem força suficiente para influenciar nos rumos tomados pelo gerenciamento do projeto.

Gerente de Projeto: possui visão mais pragmática de um projeto, voltando sua preocupação muitas vezes somente para o gerenciamento dos recursos e do tempo. Sofre vários tipos de pressões durante suas atividades, desde a exercida pelo alto escalão da organização até a do pessoal de desenvolvimento e produção. É levado muitas vezes a adotar medidas impopulares e de impacto apenas imediato no projeto.

Usuário do sistema: são os clientes diretos do sistema, cujo papel pode ser exercido por diversos atores, como operadores (técnicos e engenheiros), equipe de software, e outros participantes que também são os interessados pelos requisitos do projeto. Interagem diretamente com os produtos espaciais e tem a percepção das características ambientais, funcionais e não funcionais dos requisitos. São capazes de identificar problemas e apresentar soluções com uma rapidez muito maior que os projetistas, podendo adotar posturas contraditórias para com a utilização do sistema. São muitas vezes os responsáveis por criar e validar os requisitos de um sistema e são capazes de comprometer a integridade do sistema ou de sua operação.

5. Requisitos e Segurança

Uma forma de solucionar a falta de uma abordagem sistêmica de segurança em sistemas de software, obstáculo relacionado à visão de tecnologia, apresentado anteriormente, pode ser através da adoção de uma abordagem disciplinada de segurança dentro da atividade de elicitação. É desejável que, nesta fase, sejam realizadas atividades de extração de requisitos e análise de segurança de forma concorrente e complementar.

Segundo Kotonya e Sommerville [8], o processo de requisitos pode ser estendido para incorporar uma análise de segurança, cujo resultado será usado para alterar (quando necessário) os requisitos sugeridos para o sistema. Isto significa integrar as duas atividades, análise de requisitos e análise de segurança de forma iterativa, refinando os requisitos em cada ciclo.

No caso específico da atividade de elicitação de requisitos, a extração dos requisitos normais consiste na identificação das entidades relevantes , seu comportamento, especificação de entradas e saídas, restrições e interações com outras entidades. Já no caso particular da extração de requisitos de segurança, deve-se avaliar em todos os requisitos extraídos, potenciais perigos e riscos, e se preocupar com que a especificação não viole o comportamento seguro do sistema: considerando que requisitos normalmente especificam o que o sistema deve fazer ou espera que aconteça.

Acredita-se que uma maneira de extrair objetivamente requisitos de segurança é primeiramente classificá-los através de fatores. A proposta é tratar o assunto segurança de uma maneira mais ampla, como o enfoque do conceito de confiança no funcionamento, ou dependabilidade (dependability) [11] [22].

Seguindo a classificação de Firesmith [4], dependabilidade, é o grau com que os vários tipos de usuários, podem depender de um produto de trabalho. O autor classifica dependabilidade nos seguintes subfatores: disponibilidade operacional (operational availability), confiabilidade (reliability), robusteza (robustness) e defensabilidade (defensibility).

Disponibilidade Operacional: é o grau com que o produto do trabalho está operacional e disponível para uso num determinado instante de tempo.

Confiabilidade: é o grau com que o produto do trabalho opera sem falha sob dadas condições normais, durante um certo período de tempo.

Robusteza: é o grau com que um produto do trabalho executável continue a funcionar apropriadamente sob uma dada condição ou circunstância anormal.

Defensabilidade: é o grau com que o sistema ou componente se defende de acidentes e ataques. É classificada em Segurança contra falhas acidentais (Safety), Segurança contra falhas maliciosas/ intencionais (Security) e Capacidade de sobrevivência (Survivability).

Segurança contra falhas acidentais inclui prevenção de acidentes contra danos a saúde, propriedade e ambiente. Segurança contra falhas maliciosas/ intencionais inclui prevenção contra danos maliciosos e intencionais. Capacidade de sobrevivência: inclui prover serviços essenciais de missão crítica a despeito de danos maliciosos ou acidentais.

Segundo De Lemos [2] os requisitos devem demonstrar uma consistência entre as restrições de segurança do software e sua especificação, bem como sua completeza com respeito às propriedades de segurança. Isto significa que a técnica de extração deve ser capaz de produzir requisitos que mantenham o comportamento seguro do sistema em operação na presença de qualquer evento, condição ou circunstância. Neste sentido, após classificar os requisitos de um sistema dentro de (sub) fatores, alguns critérios podem (e devem) ser utilizados para auxiliar no detalhamento de cada requisito. Uma lista de critérios (checklist), independentemente de uma técnica especifica de elicitação, será capaz de ajudar na investigação das características e da natureza de cada requisito previamente extraído, de modo a produzir especificações mais completas, consistentes e confiáveis, principalmente com relação à segurança. Deve-se levar em consideração, por exemplo, estados, eventos, entradas e saídas, e a relação entre os disparos destes eventos e suas saídas.

6. Estudos futuros

São apresentadas nesta seção algumas abordagens e técnicas, que fazem uso tanto da engenharia de requisitos como da engenharia de projeto e que podem vir a auxiliar na solução dos problemas apresentados. A idéia é aplicar algumas destas técnicas na estratégia de elicitação de requisitos de segurança em projetos de software da área espacial.

Acredita-se que métodos como o do Volere, o do uso de alternativas de projeto (rationales) e a de organização da informação através da ontologia, possam auxiliar tanto na definição da estratégia de elicitação, bem como no próprio processo de descoberta dos requisitos de segurança de um projeto.

6.1. O Método Volere

Uma abordagem sistemática de produzir requisitos que seja capaz de conciliar as diferentes opiniões dos envolvidos e seu entendimento sobre o sistema constitui um desafio para qualquer projeto. O processo de especificação de requisitos chamado Volere [17] fornece uma estrutura e guias bem definidos sobre qual deve ser o conteúdo necessário para a identificação dos requisitos de um projeto [7].

O processo é baseado em experiência de vários projetos de análise de negócios, e está em continua melhoria. Desde sua introdução em 1995, o método Volere foi adotado por milhares de organizações em torno do mundo. O modelo proposto pelo método é composto de um guia para o conhecimento dos itens necessários a fim de se especificar os requisitos para um produto. O produto é freqüentemente parte de software, mas ele pode também ser uma parte de hardware, um produto de consumidor, um conjunto de procedimentos ou qualquer outra coisa desde que seja com o enfoque de um projeto.

O modelo conta com um checklist baseado na área de conhecimento de requisitos e de informações desta competência. O método fornece uma lista de questões que devem ser detalhadas para o levantamento dos requisitos do sistema, envolvidos no ambiente, usuários finais do produto, restrições da solução, escopo do trabalho e da estratégia, requisitos funcionais e de dados, e não funcionais, como aparência, usabilidade, desempenho, operacionais, manutenção, segurança, culturais, tarefas, custos, riscos, documentação entre outros que auxiliam no detalhamento do sistema.

Devido a diferentes culturas de engenharia, a utilização sistemática deste recurso em uma abordagem de elicitação de requisitos de segurança de software facilitará a identificação dos mesmos por parte dos diferentes tipos de participantes, podendo capturar e agrupar as diferentes visões dos envolvidos através de seu modelo de categoria dos requisitos.

A idéia é poder observar os requisitos de segurança através de uma mesma técnica, de forma organizada e categorizada, sob diferentes perspectivas dos usuários.

6.2. A Técnica Baseada em Alternativas (Rationales)

A técnica de uso de alternativas (rationales), proposta inicialmente por Maclean [13], capaz de capturar requisitos e expor os usuários a escolhas de alternativas de projeto, estimulando a utilidade e a usabilidade das escolhas. Outro benefício é a possibilidade de melhorar a comunicação de opções e encorajar um projeto mais participativo [21]. A possibilidade de soluções múltiplas, baseada em critérios de escolha, e com um enfoque na documentação do histórico das decisões e sua comunicação entre os participantes, permite explorar exponencialmente a versatilidade de um projeto.

No caso especifico de requisitos de segurança de um projeto de software para aplicação espacial, onde se trabalha com projetos de longo prazo, com repetidos ciclos de manutenções e de versões, a inclusão de uma abordagem capaz de identificar e registrar as diversas alternativas possíveis de solução, é de extrema valia.

A memória de projeto e as razões de escolha muitas vezes são perdidas no tempo e podem gerar erros sistêmicos de grandes proporções ou até mesmo catastróficos, como o do Ariane 5. Acredita-se que erros como o deste acidente espacial poderia até ser evitado se os rationales fossem revisados e discutidos em cada edição do foguete.

Outra forte contribuição do uso desta técnica é seu emprego em ambientes organizacionais heterogênicos e não muito estáveis como de uma organização militar. É possível observar – a despeito do que popularmente se acredita – que institutos governamentais de pesquisa perdem seu corpo de conhecimento com o passar dos anos. Pressões econômicas, falta de estímulos profissionais, políticas governamentais equivocadas na área tecnológica podem levar a perdas irreparáveis de conhecimento e de experiência do uso deste tipo de aplicação. Muitos pesquisadores trocam sua posição funcional estável para os desafios da iniciativa privada, ficando muitos projetos sem reter devidamente o conhecimento adquirido ao longo de sua vida. A perda dos argumentos que levaram a uma determinada escolha, ou não, a forma e os critérios definidos para captura dos requisitos são informações que podem levar até anos para serem recuperadas.

6.3. O Uso de Ontologia

A Ontologia sob a perspectiva da computação é um assunto que tem sido discutido com bastante importância na área de organização da informação, principalmente pela comunidade de pesquisa em inteligência artificial.

Como uma disciplina de filosofia, a construção de uma ontologia deve prover uma categorização de um sistema que permita construir uma certa visão de mundo em particular. O termo ontologia foi definido por Gruber [6] como sendo uma perspectiva formal e explicita de um conceito compartilhado. Ela pode ser útil para ajudar os envolvidos a compreender melhor uma área de conhecimento, dando apoio à busca de consenso no entendimento sobre um assunto específico.

Basicamente consiste de um conjunto de conceitos, relações, definições, propriedades e restrições descritas na forma de axiomas [3]. A primeira atividade a ser realizada na construção de uma ontologia é identificar claramente seu objetivo e os usos esperados para ela, isto é o domínio de interesse. Em seguida, os conceitos e as relações devem ser identificados e organizados num modelo utilizando uma linguagem gráfica. Finalmente, a Ontologia deve ser avaliada para verificar se satisfaz os requisitos estabelecidos na especificação. Esta avaliação pode ser realizada em paralelo com a captura e formalização destas propriedades, onde devem ser observados critérios como clareza, coerência, extensibilidade e compromissos ontológicos mínimos [18].

Todo o processo deve ser documentado a fim de incluir os objetivos, motivações, as descrições textuais da conceituação, formalismos e critérios de projeto.

Especificamente na atividade de elicitação de requisitos de segurança, o uso de uma ontologia para representar o domínio de conhecimento relacionado a requisitos de sistemas espaciais que contém software, acredita-se possibilitará uma descrição exata do conhecimento envolvido na área de software e sistemas espaciais, não permitindo que os requisitos elicitados tenham uma interpretação diferente do que se pretende, o que é muito comum quando do uso de linguagem natural, onde as palavras podem ter semântica diferente do contexto proposto.

Outro benefício é o compartilhamento do conhecimento e das informações do domínio da engenharia de software para aplicações espaciais, com o mínimo de interpretações dúbias, entre os diferentes envolvidos no processo. Consultas, comparações e verificações poderão ser efetuadas baseados nas aplicações de software a serem desenvolvidas.

A reutilização do conhecimento, extremamente benéfico para projetos espaciais, onde uma série de protótipos muito semelhantes pode vir a ser desenvolvido, facilitará a criação de mecanismos de inferência para criar novo conhecimento a partir do existente.

7. Considerações

Este estudo apresenta, segundo o entendimento dos autores, uma visão sobre os principais aspectos que influenciam as atividades de elicitação de requisitos em aplicações de desenvolvimento de software da área espacial.

Foram tecidos alguns comentários sobre a visão de tecnologia, pessoas e processos, os participantes no processo de elicitação e sobre como classificar a questão da segurança neste contexto. Também foram abordadas algumas técnicas que deverão ser empregadas em trabalhos futuros deste estudo.

Entende-se que uma boa prática na atividade de elicitação de requisitos significa integrar nesta disciplina diferentes aspectos como o de pessoas, processos e tecnologia. Em sistemas críticos, como o de projetos espaciais, é extremamente importante que estas visões sejam inter-relacionas de maneira a se obter uma consistência maior dos requisitos extraídos.

Na visão de tecnologia foi destacado como deve ser a condução inicial das atividades de elicitação relativa à

questão da segurança em projetos de sistemas críticos. Classificar os requisitos de acordo com fatores bem estabelecidos é o primeiro passo para desenvolver modelos consistentes e seguros. O segundo passo seria realizar uma análise de segurança em conjunto com a análise dos requisitos para a avaliação do quanto os riscos associados com a especificação são aceitáveis. É importante criar uma estratégia que combine e complemente fatores e subfatores de dependabilidade, de modo a propor soluções capazes de atender a uma especificação o mais completa e abrangente possível.

Outro aspecto importante a ressaltar, é que atualmente existem iniciativas que procuram criar novas abordagens que sejam capazes de integrar a engenharia de sistemas e a de software, como a System Modelling Language (SysML), criado pela Object Management Group (OMG), que se propõe a dar suporte a especificação, análise, projeto, verificação e validação de sistemas complexos que podem incluir hardware, software, dados, pessoas, procedimentos e facilidades. O grande benefício é criar uma linguagem comum entre times heterogêneos, como engenheiros de sistema e de software, que desenvolvem soluções de hardware e de software melhorando a comunicação entre os envolvidos.

Na visão de pessoas, os engenheiros de requisitos têm de ser integrados tanto às equipes de sistema, de software, bem como ao gerenciamento de projeto, de forma a poder obter uma visão mais abrangente e completa do problema. Esta integração deve permitir compartilhar as especialidades e os valores de cada perfil de engenharia e ainda criar uma cultura de segurança única. Deve-se alcançar um meio termo entre a aversão ao risco da engenharia de sistema versus a tendência a modismos da engenharia de software. A implantação da cultura de segurança deve propiciar um reconhecimento claro de seu valor pelos participantes, estar integrada a todas as atividades de engenharia, capaz de ser contabilizada e ter uma situação de liderança (relativamente) independente.

Na visão de processos, vários padrões se propõem a criar modelos e frameworks de desenvolvimento e avaliação de processo que estão convergindo para a integração da engenharia de sistemas e de software. Tanto o Capability Maturity Model Integration (CMMI), como as normas da European Cooperation for Space Standartization (ECSS) da ESA se preocupam com conceitos combinados de engenharia de sistemas e de software. É esperado de uma organização que desenvolve soluções para sistemas espaciais implante processos de desenvolvimento e gerenciamento compatíveis com os requisitos de qualidade espacial, e que mantenha ao longo da vida de um projeto compromisso com a melhoria constante.

É importante frisar que a estratégia de elicitação de requisitos em projetos, seja de software ou de sistema, área espacial ou não, devem empregar técnicas que visem aumentar o grau de confiança de que estes são os requisitos corretos, completos e adequados para a solução da solicitação proposta.

O enfoque foi direcionado para a fase de requisitos de um sistema que contém software, mas especificamente as atividades de elicitação. Ferramentas, técnicas e abordagem, tanto formais, como não formais devem ser estudadas e criticadas a fim de se certificar qual melhor se encaixa no domínio do problema, da solução e da equipe escolhida para desenvolver tal projeto. Podemos observar que as abordagens Volere, de alternativas e de ontologicas, em uma primeira aproximação, contribuirão extensivamente para a definição do escopo do projeto e de sua solução.

Concluindo, em sistemas espaciais, especialmente aqueles que possuem software como elemento crítico, precisam adotar uma nova estratégia para superar as dificuldades relacionadas a problemas com requisitos e projeto.

Os obstáculos relacionados à tecnologia e processos no âmbito do programa espacial brasileiro, devem ser focados principalmente na mudança de cultura das pessoas envolvidas. Em qualquer que seja a fase do projeto ou a disciplina envolvida, esta mudança deve ser calcada na qualidade, e mais especificamente no fator de segurança.

12. Referências

[1] I M. Almeida, C. W. Johnson: Extending the Borders of Accident Investigation: Applying Novel Analysis Techniques to the Loss of the Brazilian Space Programme's Launch Vehicle VLS-1 V03, 2005, http://www.dcs.gla.ac.uk/~johnson/papers/papers.html.

[2] R. De Lemos: Requirements Engineering for Embedded Systems, tutorial, Centro Técnico Aeroespacial. São José dos Campos, 1998.

[3] K. Duarte; R. A. Falbo, Uma Ontologia de Qualidade de Software, Anais do VII Workshop de Qualidade de Software, XIV Simpósio Brasileiro de Engenharia de Software, João Pessoa; 2000.

[4] D. G. Firesmith, Engineering Safety Requirements, Safety Constraints, and Safety-Critical Requirements, Journal of Object Technology, Zurich, Switzerland, March/April, 2004, pp 27-42.

[5] R. Gonzales, Developing the Requirements Discipline: Software vs. Systems, IEEE Software, March/April, 2005, pp 59-61.

[6] T. Gruber, A translation Approach to Portable Ontology Specification, 1993.

[7] D.T. Haley, B. Nuseibeh, H. C. Sharp, J. Taylor, The Condrum of Categorizing Requirements: Managing

Requirements for Learning on the Move, 12th IEEE International Requirements Engineering Conference, 2004.

[8] G. Kotonya, I. Sommerville, Requirements Engineering, John Wiley & Son Ltd (2000).

[9] C. H. N. Lahoz, J. B. Camargo Jr., M. A D. Abdala, L. A. Burgareli, A Software Safety Requirements Elicitationn Study on Critical Computer Systems. 1st IET International Conference on System Safety, London, UK, 2006.

[10] P. A Laplante, Real Time Systems Design and Analysis, IEEE Press Wiley- Interscience, John Wiley & Sons, 2004.

[11] J.-C. Laprie, Dependable Computing: Concepts, Limits, Challenges, Special Issue of the 25th International Symposium on Fault-Tolerant Computing, Pasadena, USA. IEEE Computer Society Press. Los Alamitos, CA., 1995, pp 42-54.

[12] J.L. Lyons, Report of the inquiry board into the failure of Flight 501 of the Ariane 5 rocket.Technical report, European Space Agency, Paris, France, 1996.

[13] A. Maclean, R. M. Yong, V Belloti, T. Moran, Questions, Options, and Criteria: Elements of Design Space Analysis, Human-Computer Interaction 6(3&4), 1991.

[14] NASA, Mars Climate Orbiter: Mishap Investigation Board, Phase I Report, Technical report, Mars Climate Orbiter, Mishap Investigation Board, NASA Headquarters, Washington DC, USA, 1999, ftp://ftp.hq.nasa.gov/pub/pao/reports/1999/MCO report.pdf.

[15] NASA/JPL, Report on the loss of the Mars Polar Lander and Deep Space 2 Missions, Technical Report JPL D-18709, NASA/Jet Propulsion Laboratory, California Institute of Technology, 2000, http://www.jpl.nasa.gov/marsreports/marsreports.html.

[16] A. Saeed, R. De Lemos, T. Anderson, The Role of Formal Methods in the Requirements Analysis of Safety-Critical Systems: a Train Set Example, Proceeding of the 21st Symposium on Fault Tolerant Computing, Montreal, Canada, 1991, pp 478-485.

[17] S. Robertson and J. Robertson, Mastering the Requirements Process, Harlow, England: Addison-Wesley, 1999.

[18] A. I. S.C.C. Santos, F. Hustinx, N. A Rosário, P. F. M. Pinto, Ontologia, Seminário, Faculdade de Engenharia da Universidade do Porto, 2005.

[19] P. Sayer, G. Kotonya, SWEBOK, Software Requirements Engineering Knowledge Area Description, Technical Version 5, 1999,http://www.swebok.org/stoneman/version_0.5/KA_Description_for_Software_Requirements_Analysis(Version_0_5).pdf.

[20] Serviço Público Federal. Ministério da Defesa, Comando da Aeronáutica, Departamento de Pesquisas e Desenvolvimento, Relatório da Investigação do acidente ocorrido com o VLS1-V03, em 22 de agosto de 2003, em Alcântara, Maranhão, São José dos Campos, Brasil, 2004, http://www.aeb.gov.br.

[21] A. Sutcliffe, Requirements Rationales: Integrating Approaches to Requirements Analysis, ACM, 1995.

[22] P. Veríssimo, R. De Lemos, Confiança no Funcionamento: Proposta para uma Terminologia em Português. Publicação conjunta INESC e LCMI/UFSC, 1989, http://www.cs.kent.ac.uk/people/staff/rdl/CoF/.