



Pós-Graduação em Ciência da Computação

ABORDAGEM DA ENGENHARIA DE REQUISITOS EM PROJETOS DE DESENVOLVIMENTO DE SOFTWARE PARA TELESSAÚDE/TELEMEDICINA

ALEXANDRE JOSÉ HENRIQUE DE OLIVEIRA LUNA



Universidade Federal de Pernambuco
Centro de Informática

posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE, JULHO/2008

ALEXANDRE JOSÉ HENRIQUE DE OLIVEIRA LUNA

**ABORDAGEM DA ENGENHARIA DE REQUISITOS EM PROJETOS DE
DESENVOLVIMENTO DE SOFTWARE PARA
TELESSAÚDE/TELEMEDICINA**

Monografia apresentada ao Curso de pós-graduação em Ciência da Computação, Centro de Informática, Universidade Federal de Pernambuco, como requisito parcial para conclusão da disciplina Engenharia de Requisitos.

Professor: Prof. Ph.D Jaelson Brelaz de Castro

Recife
15 de Julho de 2008

RESUMO

Em virtude dos avanços das tecnologias da informação e das telecomunicações, a Telemedicina e a Telessaúde, bem como outras áreas de pesquisa e desenvolvimento da Informática Médica, tomaram um grande impulso. A Telemedicina pode ser definida como o uso das tecnologias da informação e de comunicações para apoiar a assistência médica à pacientes em locais distantes, baseando-se na troca de informações. Já a Telessaúde pode ser definida como o uso da tecnologia da informação e comunicação para prover serviços de saúde, conhecimento em informação nos contextos onde a distância é um fator crítico. Os termos se confundem em algumas abordagens, mas são melhores descritos como: complementares.

A Telemedicina e a Telessaúde podem ser realizadas com o que existe de mais moderno e sofisticado em termos de tecnologias, porém os custos de implantação e de manutenção destes sistemas podem muitas vezes inviabilizar a realização desta atividade na rotina médica. Por esta razão, além de ter um custo acessível, um sistema de Telemedicina precisa também ser cuidadosamente planejado e projetado.

Neste contexto os processos de Engenharia de Requisitos são extremamente importantes para minimizar os riscos do ciclo de desenvolvimento de projetos desta natureza.

Sob esta ótica estaremos estudando o estado da arte da abordagem de Engenharia de Requisitos em projetos de desenvolvimento de software nas áreas de telessaúde e telemedicina, pesquisando trabalhos em andamentos em grupos de pesquisas distintos e apresentando o confronto destas visões em comparação as experiências recentemente vivenciadas pelo autor em um caso prático.

Palavras-chave: Gerenciamento de Projetos, Engenharia de Software, Engenharia de Requisitos, Telessaúde, Telemedicina

ABSTRACT

Because of advances in information technology and telecommunications, telemedicine and Telehealth, and other areas of research and development of Medical Informatics, took a big boost. The Telemedicine can be defined as the use of information technology and communications to support medical care to patients in remote locations, relying on the exchange of information. Already the Telehealth can be defined as the use of information technology and communication to provide health services, knowledge on information in contexts where the distance is a critical factor. The terms are confused in some approaches, but are best described as: complementary.

The Telemedicine and Telehealth can be made with what exists in most modern and sophisticated in terms of technology, but the costs of establishing and maintaining these systems can often prevent the realization of this activity in the routine medical. For this reason, besides having an affordable, a system of telemedicine must also be carefully planned and designed.

In this context the processes of engineering requirements are extremely important to minimize the risks of the cycle of development projects of this nature.

From this perspective we are studying the state of the art of the approach of Engineering Requirements in developing software in the areas of Telehealth and telemedicine, searching work in progress in different groups of searches and presenting the confrontation of these visions compared the experiences recently experienced by author in a case study.

Keywords: Project Management, Software Engineering, Requirements Engineering, Telehealth, Telemedicine

LISTA DE FIGURAS

Figura 1- SCRUM work flow diagram. Fonte: adaptado de The Scrum Development Process [31].	25
Figura 2- O processo da Engenharia de Requisitos: entradas e saídas. Fonte: SOMMERVILLE [22].	29
Figura 3- Modelo Espiral aplicado à Engenharia de Requisitos. Fonte: SOMMERVILLE [22].	31
Figura 4-Estrutura de um documento de requisitos. Fonte: IEEE [24]	34
Figura 5 - Processo de elicitação de requisitos. Fonte: SOMMERVILLE [22].	38
Figura 6 – Esquema do Método Volere. Fonte: Adaptado de Robertson [41].	63
Figura 7 – Formulário de solicitação de biópsia e laudo no Sistema HealthNet. Fonte: Mello [37].	65
Figura 8 – Telas do Módulo de Biblioteca Multimídia do dataNUTES - (a) Resultados do sistema de busca por palavras-chave; (b) O módulo Desktop realize o upload da mídia, a ser classificada e publicada na Web, para o servidor de Streaming do dataNutes; (c) Exibição do vídeo em modo Streaming, através do Player Web. Fonte: LUNA [5].	70

SUMÁRIO

1	Introdução	8
1.1	Contexto e Motivação.....	8
1.2	Objetivos.....	11
1.3	Organização do Trabalho.....	11
2	Bases Conceituais	14
2.1	Informática Médica.....	14
2.1.1	Telessaúde.....	15
2.1.2	Telemedicina.....	16
2.1.3	Teleconferência.....	18
2.1.4	Teleconsulta.....	20
2.2	A Engenharia de software.....	21
2.2.1	Métodos Ágeis: SCRUM.....	24
2.2.2	A Engenharia de Requisitos.....	26
2.2.2.1	O processo de Engenharia de Requisitos.....	28
3	Engenharia de Requisitos aplicada à Saúde	37
3.1	Técnicas para Elicitação de Requisitos.....	39
3.2	Técnicas de Modelagem de Requisitos.....	44
3.3	Técnicas de Validação de Requisitos.....	47
3.3.1	Revisão de Requisitos.....	49
4	Particularidades da abordagem da Engenharia de Requisitos para softwares aplicados à Saúde	52
4.1	Aspectos que influenciam a aplicação de Engenharia de Requisitos no domínio da Saúde.....	52
4.1.1	O lado humano (Stakeholders).....	53
4.1.2	Vários pontos de vista: um mesmo problema.....	57
4.1.3	O grande encontro: homem e software.....	59
4.2	Algumas abordagens de Engenharia de Requisitos para Projetos de Telessaúde/ Telemedicina.....	62
4.2.1	Método Volere para Estudo de Requisitos em Softwares Embarcados para Telemedicina.....	62
4.2.2	Cenários, Casos de Uso e UML para desenvolvimento de um Sistema Integrado de apoio ao Telediagnóstico e à Segunda Opinião Médica: HealthNet.....	65
4.2.3	Prototipação Evolutiva e SCRUM para desenvolvimento do Módulo de Biblioteca Multimídia do dataNUTES: uma experiência de desenvolvimento distribuído de software.....	67
5	Conclusão e considerações finais	74
6	Referências	77

Capítulo 1

1 Introdução

Neste capítulo, constaram as principais questões que motivaram a realização deste trabalho, o objetivo da pesquisa e a organização da monografia.

1.1 Contexto e Motivação

Segundo a Organização Mundial da Saúde (OMS), Telemedicina é o uso da tecnologia da informação para entregar serviços e informações médicas de um local para o outro. Nos casos onde a distância é um fator crítico, tais serviços são promovidos por profissionais da área de saúde, usando tecnologias de informação e comunicação para o intercâmbio de informações válidas para diagnósticos, prevenção e tratamento de doenças, e a contínua educação de profissionais de saúde, assim como para fins de pesquisas e avaliações [1]. Num país de dimensões continentais, como o Brasil, e com oferta assimétrica de serviços na área de Saúde, a Telemedicina pode ser uma oportunidade/opção de resolução de dificuldade de comunicação entre instituições de saúde, garantindo à população uma maior qualidade na oferta de serviços e benefícios do sistema nacional de saúde [2].

A Telemedicina e a Telessaúde podem ser realizadas com o que existe de mais moderno e sofisticado em termos de tecnologias, porém os custos de implantação e de manutenção destes sistemas podem muitas vezes inviabilizar a realização desta atividade na rotina médica. Por esta razão, além de ter um custo acessível, um sistema de Telemedicina precisa também ser cuidadosamente planejado e projetado. Neste contexto os processos de Engenharia de Requisitos são extremamente importantes para minimizar os riscos do ciclo de desenvolvimento de projetos desta natureza.

Um estudo revelou que cerca de 37% dos problemas que ameaçam os projetos estão relacionados com requisitos, fazendo deste tópico o mais problemático dentre os diversos fatores levantados [3].

Portanto, o estudo dos requisitos para o correto desenvolvimento de qualquer tipo de software é de grande importância para o sucesso do projeto. Para o desenvolvimento de softwares aplicados a saúde, o entendimento não correto por parte do analista de sistemas ou a expressão não clara e ambígua do cliente, podem ser combinações perigosas a saúde das pessoas. Dessa forma, torna-se um trabalho bastante delicado e minucioso da elicitación até a correta implementação do requisito em um software de saúde [4].

O Grupo de Tecnologias de Informação em Saúde (TIS), entidade de pesquisa vinculada ao Núcleo de Telessaúde do Hospital das Clínicas (NUTES/HC) da Universidade Federal de Pernambuco (UFPE) ao longo de anos vem desenvolvendo projetos de pesquisa com foco na Telessaúde/Telemedicina. Dentre eles, merece destaque o projeto de Biblioteca Multimídia do Sistema de Gestão de Serviços de Telessaúde (dataNUTES). Este tem como objetivos disponibilizar uma biblioteca de sessões de videoconferências na Web, tendo como estudo de caso as sessões produzidas pela Rede de Núcleos de Telessaúde (NUTES) de Pernambuco, construir um sistema de armazenamento eficiente e de boa qualidade para essas sessões e ainda, elaborar uma biblioteca de imagens médicas, utilizando a internet como meio para disponibilização [32].

A Rede NUTES [35] é resultado do projeto de Telessaúde no Programa de Saúde da Família (PSF), um dos projetos do Grupo TIS, que visa disponibilizar instrumentos de Telemedicina/Telessaúde como uma alternativa para melhorar o acesso à saúde para a população carente, em zonas rurais ou desprovidas de serviços médicos especializados, promovendo o cuidado à saúde através de programas de educação e prevenção ao nível da atenção primária. Este Projeto piloto foi financiado pelo Ministério da Saúde, através

da UFPE. É utilizado para esses fins dois serviços de telessaúde: i) sistema de cooperação HealthNet pela Internet; ii) programas de capacitação para o PSF através de videoconferências.

O projeto BIBLIOTECA MULTIMÍDIA dataNUTES (BMD) é subdividido em três subprojetos. O primeiro subprojeto, BMD-Videoconference, tem como foco disponibilizar o conteúdo gerado através das videoconferências através de outros canais de comunicação, no caso, a Internet. Este conteúdo não se desatualiza rapidamente podendo ser utilizado para futuras capacitações e reciclagens, ou mesmo para profissionais que não estavam presentes nas sessões realizadas. Utilizando a Internet é possível disponibilizar esses vídeos para os mais distantes usuários superando limitações geográficas e logísticas. O segundo subprojeto, BMD-DVD, visa elaborar um sistema digital para armazenar essas sessões de videoconferências, uma vez que o armazenamento analógico (fita VHS) limita a disponibilização do conteúdo gerado para os profissionais de saúde, bem como prejudica a qualidade das sessões geradas. O terceiro subprojeto, BMD-Radiologia, vem a trabalhar com imagens médicas (raios-X, tomografias computadorizadas, ultra-sons, dentre outras). Ele tem como objetivo criar uma biblioteca de imagens médicas oriundas do serviço de radiologia do Hospital das Clínicas da UFPE e do sistema HealthNet, utilizando a internet como meio difusor, onde neste sistema profissionais e estudantes da área da saúde, poderiam ter acesso para discussão de casos, como também para o aprendizado acadêmico [33].

Através do Projeto BIBLIOTECA MULTIMÍDIA dataNUTES (BMD) viabilizamos os princípios tecnológicos e implementamos o processo de armazenamento, conversão e disponibilização em ambiente Web os conteúdos multimídia produzidos pelo NUTES/HC, aplicáveis ao BMD-Videoconference e ao BMD-Radiologia [5].

O Projeto BIBLIOTECA MULTIMÍDIA dataNUTES (BMD), experiência recentemente vivenciada pelo autor, será utilizado neste trabalho, como caso prático para o estudo,

confronto e comparação das visões encontradas entre as pesquisas de trabalhos em andamento em grupos de pesquisas distintos, na abordagem de Engenharia de Requisitos em projetos de desenvolvimento de software nas áreas de telessaúde e telemedicina.

1.2 Objetivos

Neste trabalho faremos uma revisão sobre os aspectos envolvidos nos processos de Engenharia de Requisitos para projetos de desenvolvimento de software nas áreas de telessaúde e telemedicina, dando ênfase ao processo de elicitação de requisitos, com suas técnicas, práticas, além dos cuidados necessários para o correto desenvolvimento de softwares para aplicação em áreas relacionadas à saúde humana.

1.3 Organização do Trabalho

Além desta seção introdutória que é o capítulo 1, a monografia está dividida em mais quatro capítulos, como segue:

O **capítulo 2** apresenta o embasamento teórico dos conceitos relacionados aos temas abordados por este trabalho.

No **capítulo 3** encontra-se uma discussão entre as diversas técnicas de elicitação e validação e a que melhor se adequa ao tipo de software em questão, baseado na experiência prática do autor.

No **capítulo 4** faremos uma análise de aspectos que influenciam fortemente a abordagem da Engenharia de Requisitos para Projetos de Desenvolvimento de Softwares

Alexandre José Henrique de Oliveira Luna

para Telessaúde/ Telemedicina; apresentaremos a discussão de algumas abordagens de experiências de trabalhos em andamentos em grupos de pesquisas distintos, apresentando o confronto destas visões em comparação das experiências recentemente vivenciadas pelo autor neste contexto.

O **capítulo 5** terá as considerações finais e conclusões para uma abordagem eficaz de engenharia de requisitos no desenvolvimento de softwares para esta área de aplicação, baseadas em caso prático, vivenciado pelo autor.

Capítulo 2

2 Bases Conceituais

2.1 Informática Médica

A Informática Médica ou Informática em Saúde (em Inglês Medical Informatics) é definida por Blois e Shortliffe [10] como "um campo de rápido desenvolvimento científico que lida com armazenamento, recuperação e uso da informação, dados e conhecimento biomédicos para a resolução de problemas e tomada de decisão".

Outra definição bastante difundida passa pela aprovação da Sociedade Brasileira de Informática em Saúde [11]:

"A Saúde é uma das áreas onde há maior necessidade de informação para a tomada de decisões. A Informática Médica é o campo científico que lida com recursos, dispositivos e métodos para otimizar o armazenamento, recuperação e gerenciamento de informações biomédicas. O crescimento da Informática Médica como uma disciplina deve-se, em grande parte: aos avanços nas tecnologias de computação e comunicação, à crescente convicção de que o conhecimento médico e as informações sobre os pacientes são ingerenciáveis por métodos tradicionais baseados em papel, e devido à certeza de que os processos de acesso ao conhecimento e tomada de decisão desempenham papel central na Medicina moderna".

Podemos destacar como suas principais áreas de atuação:

- Sistemas de Informação em Saúde
- Prontuário Eletrônico do Paciente
- Telemedicina

- Sistemas de Apoio à Decisão
- Processamento de sinais biológicos
- Processamento de Imagens Médicas
- Internet em Saúde
- Padronização da Informação em Saúde

2.1.1 Telessaúde

Apesar de encontrarmos diversas referências na Internet às duas palavras, nenhuma das duas já foi inserida como verbete nos dicionários procurados (Dicionário Houaiss da Língua Portuguesa e Novo Dicionário Aurélio – Século XXI). Ao analisarmos a estrutura da palavra, podemos notar que ela seria apenas uma composição por justaposição do prefixo “tele” + substantivo “saúde”. Encontramos no Houaiss [8] uma referência ao uso da palavra tele para composição de palavras:

Tele (elemento de composição) antepositivo, do adv. gr. tēle 'longe, ao longe, de longe'; ocorre em vários cultismos do século XIX em diante, com as segg. acp.: 1) 'longe, à distância': telégrafo, telegrama, telemetro, telepatia; 2) 'telegráfico (ou radiotelegráfico)': telefotografia, telemecânica; 3) 'televisão': teledrama, telenovela, telerreportagem, telespectador.

Como pudemos ver, mantém-se o som original da palavra associada a tele, mesmo que seja necessária a duplicação da letra inicial (se esta for ‘r’ ou ‘s’). Exemplo: telerreportagem. Pelo Aurélio [9] ainda temos: telerradiografia, telerrobô, telessorteio. Diante disso escolhemos utilizar o termo Telessaúde, que está de total acordo com a regra formal da língua portuguesa.

Já a University Health Network - UHN, até 1999 conhecida como Toronto Hospital, define [12]:

Alexandre José Henrique de Oliveira Luna

Telehealth is the use of information and communication technologies to provide health care when provider and recipient are separated by distance.

De acordo com o Health Telematics Unit da University of Calgary (CAN) [13]:

Telehealth is the use of information and communication technology (ICT) to deliver health services, expertise and information over distance. It includes Internet or web-based “e-health” and video-based applications, and can be delivered real-time (live) or through store-and-forward (record now, view later) mode. Telehealth is changing the way we think about health and provide healthcare in Canada. It supports innovations in health, as well as in healthcare delivery, health education, and distribution of health related information such as medical records. Telehealth is adding value. Telehealth is also changing the way we think about health and conduct healthcare around the world. It is a unique global tool which has the capability to cross all existing geographical, temporal, political, social and cultural barriers within the health sector. Telehealth will impact us all.

2.1.2 Telemedicina

Embora o tema pareça tão moderno, é bastante antigo, se desvincularmos o uso da informática como mediador da comunicação. Sabe-se da utilização do telefone para auxílio ao diagnóstico desde 1897; a transmissão de imagens de radiografias por meio telefônico foi feita na década de 1940. Apesar disso, o custo da transmissão de dados em velocidade adequada fez com que as tentativas de utilização da telemedicina viessem ser malsucedidas até a década de 1980. O surgimento de microcomputadores de mesa com sistemas de uso simples e alta capacidade computacional a custos acessíveis para o usuário individual também contribuíram para as primeiras experiências factíveis. Mais amplamente podemos conceituar a telemedicina como a combinação das tecnologias de

informática, robótica e telecomunicações com a proficiência médica, provendo condições de enviar e receber informações e realizar procedimentos.

Essa combinação objetiva viabiliza ações médicas em que os profissionais e pacientes não estão fisicamente e/ou temporalmente próximos. Outro aspecto importante da telemedicina é a possibilidade de se levar o conhecimento de especialistas a outros médicos, beneficiando a eles e seus pacientes. Médicos brasileiros podem nivelar/compartilhar seus conhecimentos com os de seus colegas de outros países. Como alternativa aos congressos e jornadas presenciais, o processo de reciclagem médica por telemedicina oferece, com custo muito mais baixo, a possibilidade de atingir grande número de profissionais, sem a necessidade de deslocamento, e minimizando os custos envolvidos.

O termo “telemedicina” [14] restringe-se à Medicina, às atividades dos médicos e “telessaúde” é mais abrangente, pois inclui todos os profissionais e atividades relacionadas com a saúde: enfermagem, odontologia, saúde pública, etc.

De acordo com Heloísa Melo Hertzog [15], “telemedicina” é a oferta de serviços ligados aos cuidados com a saúde nos casos em que a distância é um fator crítico. Tais serviços são providos por profissionais da área de saúde, usando tecnologias de informação e de comunicação para o intercâmbio de informações válidas para diagnósticos, prevenção e tratamento de doenças e a contínua educação dos provedores de cuidados com a saúde, assim como para fins de pesquisas e avaliações; tudo no interesse de melhorar a saúde das pessoas e de suas comunidades.

De acordo com György Böhm [16], professor da Faculdade de Medicina da Universidade de São Paulo - FMUSP, a “telemedicina” não é uma especialidade nova, mas um novo meio, talvez irrecusável, de se praticar a medicina de sempre. É uma tecnologia a serviço da prática médica.

De acordo com Chao Lung Wen [16], também professor da FMUSP, a “telemedicina” envolve o uso de tecnologias da informação modernas, especialmente recursos de telecomunicação e informática, a fim de prover o atendimento a pacientes que ocupam áreas de difícil acesso e de facilitar a troca de informações entre profissionais de saúde e médicos especialistas que se encontram fisicamente distantes entre si.

De acordo com o British Medical Journals [17]:

Telemedicine is an umbrella term that encompasses any medical activity involving an element of distance. In its commonly understood sense, in which a doctor-patient interaction involves telecommunication, it goes back at least to the use of ship to shore radio for giving medical advice to sea captains. A few years ago the term telemedicine began to be supplanted by the term Telehealth, which was thought to be more "politically correct," but in the past year or so this too has been overtaken by even more fashionable terms such as online health and e-health.

Segundo o Emergency Medicine and Primary Care [18], ainda pode ser vista como:

Telemedicine is the interactive audiovisual communication between health care providers and their patients. The first use of telemedicine was in 1959 when x-ray images were transmitted across telephone lines. Since that time the technology has advanced, making telemedicine more affordable and clinically useful.

2.1.3 Teleconferência

De acordo com o conselho brasileiro de Telemedicina e Telessaúde [14], no contexto da telemedicina, o termo designa uma busca de esclarecimento diagnóstico ou

orientação terapêutica (uma segunda opinião), pelo médico e seu paciente, de um profissional ou instituição mais experiente, só alcançável por telecomunicação.

Teleconferências de texto, onde as mensagens de todos os participantes são exibidas automaticamente em uma área comum, que aparece na tela de cada um como um quadro branco, exclusivamente em texto, ou, em alguns casos, podendo ser enviadas imagens. Esta é a forma de interação multilateral síncrona mais comum, e historicamente a primeira a ser desenvolvida, com base em um protocolo (uma espécie de convenção ou padrão de comunicação) denominada IRC (Internet Relay Chat). Um programa muito utilizado pelos adeptos desse formato se chama mIRC. Outro serviço de grande penetração mundial é o ICQ (pronunciado "I seek you", ou "eu procuro você"), que também exige um software especial. Ambos são disponíveis gratuitamente. O tipo mais usado atualmente, entretanto, é o "bate-papo" de texto via WWW. Já se usa esse recurso para realizar reuniões virtuais em todo o mundo, como, por exemplo, as reuniões clínicas mensais realizadas pela Sociedade de Cardiologia do Estado de São Paulo, da qual participam dezenas de médicos de todo o Brasil, e que pode compartilhar imagens.

Teleconferências de áudio e/ou vídeo, que permitem a transmissão de voz e de imagens dos participantes, que são captadas em seus microcomputadores através de microfones e pequenas câmaras de vídeo (chamadas de Webcams) que ficam geralmente em cima do monitor. É grande o número de programas e padrões existentes para implementar esse tipo de "bate-papo". Os mais conhecidos são o CUSeeMe, desenvolvido pela Universidade Cornell, dos EUA (pronuncia-se "see you, see me"), o Netscape Conference e o Microsoft NetMeeting. Os programas mais modernos combinam as funções de áudio e vídeo com um "bate-papo" de texto, transferência de arquivos entre os participantes (envio de um software, por exemplo), "passeio" colaborativo pela Internet, e exibição de desenhos e imagens ("quadro branco").

Televideoconferências: Quando utiliza vídeo e áudio interativo bi- ou multilateral, temos um tipo especializado de teleconferência, que se chama videoconferência, ou "colaboração visual", um novo nome, mais abrangente, pois funciona como um canal de TV bidirecional (como é usado todo o tempo pelas emissoras) e proporciona uma grande naturalidade à colaboração entre essas pessoas.

2.1.4 Teleconsulta

Abrange aplicações onde através de mecanismos eletrônicos como formulários da WWW um paciente pode passar informações a um médico e esse poderá realizar então o diagnóstico. Esses tipos de consultas sem a presença física não são regulamentados em nosso país e mesmo no exterior, e o médico não pode prescrever um tratamento a partir de uma teleconsulta; desse modo, o desdobramento mais comum da teleconsulta é o da segunda opinião, onde um médico em uma consulta presencial coleta os dados e solicita o auxílio de um especialista ou grupo de especialistas, ou mesmo de uma equipe mais experiente, e esses por sua vez respondem ao médico, que aplicará a terapia de forma convencional, ou seja, numa consulta presencial. Os termos comumente encontrados, mas que são na verdade derivações da teleconsulta são telediagnóstico e teleaconselhamento.

De acordo com o conselho brasileiro de Telemedicina e Telessaúde [14], teleconsulta é o processo pelo qual se realiza uma consulta médica à distância. Para isso, pode ser empregado qualquer meio tecnológico que transporte som, imagem ou comunicação escrita. O conceito essencial é que não há contato presencial entre quem faz o ato médico e quem o recebe.

Para a generalidade das especialidades médicas as consultas podem ser feitas por Teleconsulta. O objetivo de um sistema de Teleconsulta é a transmissão de som e imagem em tempo real para um outro local recorrendo a uma infra-estrutura de

telecomunicações. A imagem pode ser captada através de câmaras vulgares, câmaras de documentos ou diretamente a partir de equipamentos que produzam imagem.

Uma interação onde o paciente consulta diretamente o médico, utilizando qualquer forma de telecomunicação, incluindo a Internet. A “teleconsulta” ou consulta em conexão direta, onde não há uma presente relação médico-paciente nem exames clínicos, e onde não há um segundo médico no mesmo lugar, cria certos riscos. Por exemplo, incerteza relativa à confiança, confidencialidade e segurança da informação intercambiada, assim como a identidade e credenciais do médico.

2.2 A Engenharia de software

Engenharia de software é uma área do conhecimento da informática voltada para a especificação, desenvolvimento e manutenção de sistemas de software aplicando tecnologias e práticas de ciência da computação, gerência de projetos e outras disciplinas, objetivando organização, produtividade e qualidade [28].

Pressman [28] destaca que a Engenharia de software abrange três componentes básicos:

1. **Métodos:** proporcionam os detalhes de como construir o software. Englobam tarefas como planejamento e estimativa de projeto, análise de requisitos de software e de sistemas, projeto da estrutura de dados, arquitetura de programa e algoritmo de processamento, codificação, teste e manutenção;
2. **Ferramentas:** existem para sustentar cada um dos métodos. São muito usadas ferramentas CASE;

3. **Procedimentos:** constituem o elo entre métodos e ferramentas. Definem a seqüência em que os métodos são aplicados.

Atualmente, essas tecnologias e práticas englobam linguagens de programação, bases de dados, ferramentas, plataformas, bibliotecas, padrões, processos e a questão da Qualidade de Software.

Os fundamentos científicos para a engenharia de software envolvem o uso de modelos abstratos e precisos que permitem ao engenheiro especificar, projetar, implementar e manter sistemas de software, avaliando e garantindo suas qualidades. Além disso, a engenharia de software deve oferecer mecanismos para se planejar e gerenciar o processo de desenvolvimento de um sistema de informação.

Desenvolver software não é uma tarefa trivial, as mudanças no mundo ocorrem cada vez mais rápidas e cada vez com mais freqüência, os produtos que consumimos estão mais globalizados, mais internacionalizados. E essas mudanças afetam diretamente os softwares que tendem a atingir mais e mais usuários em lugares distintos [6].

E inegavelmente os softwares aplicados a saúde também são reflexos desse fenômeno, sofrem e ganham com as mudanças do mundo globalizado. Ganham por não possuírem fronteiras, nacionalidade, sofrem por cada vez mais necessitarem de robustez, de integração, de interoperação, de flexibilidade.

Segundo a ISO (*International Standards Organization*) [19], padrão é um documento estabelecido por consenso e aprovado por um grupo reconhecido, que estabelece um conjunto de regras, protocolos ou características de processos com o objetivo de ordenar e organizar atividades em contextos específicos, para o benefício de todos.

Para que a engenharia de software funcione em sua plenitude, temos que garantir que os requisitos do software em questão sejam corretos, bem elicitados, desenvolvidos, compreendidos pelos desenvolvedores, analistas e clientes.

Para desenvolver software de qualidade, antes de tudo temos que ter pessoas capacitadas disponíveis e um ambiente propício ao desenvolvimento. A partir desse ponto surgirão os problemas inerentes ao desenvolvimento do software em si. Questões como restrições a aspectos legais, restrições de hardware, restrições de tempo e de pessoas dentre outras, e essas restrições devem ser cuidadosamente observadas durante todo o processo de desenvolvimento.

As dificuldades no desenvolvimento destes artefatos de software são inúmeras. Vão desde a primeira estimativa irreal do desenvolvimento do projeto até a especificação de uma solução mal concebida, desnivelando indicadores de eficiência, de risco de sucesso, de satisfação do usuário [28], entre outros.

Segundo Kondo [29], há assim um consenso entre aqueles que estão encabeçando projetos desta natureza: a necessidade de uma análise de requisitos mais intensa, mergulhando nos problemas e validando todo entendimento dos processos de negócio envolvidos com os stakeholders. Podemos destacar as seguintes dificuldades computacionais:

- **Relativas ao tempo de resposta das interações com a aplicação:** uma das características essenciais do software de Telemedicina é o pronto feedback dado ao médico, ao pesquisador, ou a qualquer agente de saúde;
- **Relativa à comunicação humano-computador [30]:** o computador possui um padrão próprio em seu protocolo de comunicação, nem sempre apresenta baixa complexidade de uso;

- **Relativas à expansão de tecnologias de telecomunicação:** robusta e apropriada, agregada ao baixo custo;
- **Em adicionar dados na base de conhecimento do paciente:** a construção do conhecimento baseia-se no conhecimento pré-adquirido do paciente para atender um novo assunto

2.2.1 Métodos Ágeis: SCRUM

O SCRUM é uma abordagem para desenvolvimento de software. O termo SCRUM vem de um estudo feito por Takeuchi e Nonaka [26]. Como resultado deste estudo, foi percebido que projetos usando equipes pequenas e multidisciplinares (cross-functional) produzem os melhores resultados, como foi o caso do Projeto Biblioteca Multimídia dataNUTES [5].

De acordo com Schwaber [27], um projeto desenvolvido a partir do SCRUM se inicia com uma visão do produto (solução) a ser desenvolvido, contendo a lista das características, premissas e restrições estabelecidas pelo cliente, chamada Product Backlog, contendo todos os requisitos conhecidos, uma vez identificadas as suas prioridades, são divididas nos diversos releases.

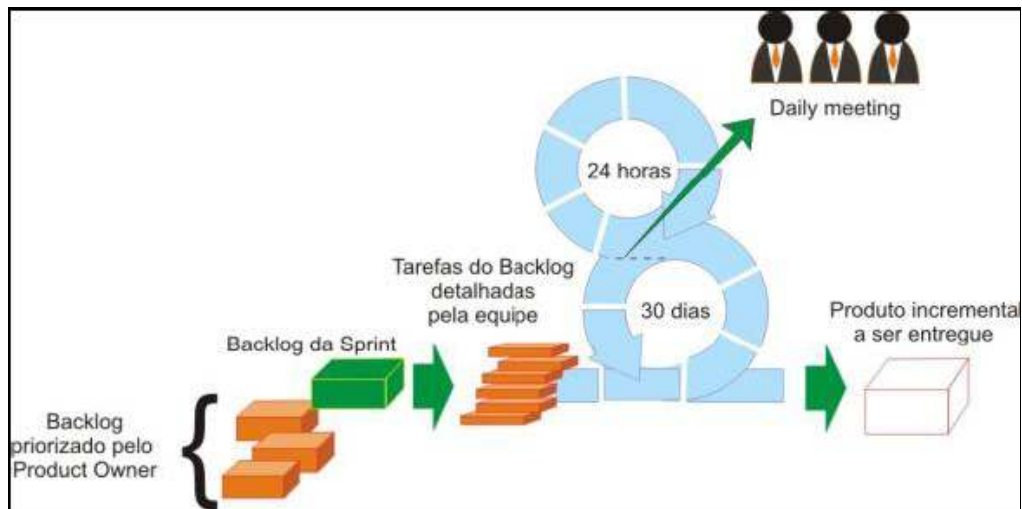


Figura 1- SCRUM work flow diagram. Fonte: adaptado de The Scrum Development Process [31].

No SCRUM, são realizadas iterações chamadas de Sprints. Schwaber explica que cada Sprint começa com uma reunião de planejamento (Sprint Planning Meeting), na qual o Product Owner e o Time decidem em conjunto o que deverá ser implementado (Selected Product Backlog). A reunião é dividida em duas partes. Na primeira parte (Sprint Planning 1), o Product Owner apresenta os requisitos de maior valor e prioriza aqueles que devem ser implementados. A lista de tarefas pode ser modificada ao longo da Sprint pelo Time e as tarefas podem variar entre 4 a 16 horas para a sua conclusão.

Na execução das Sprints, diariamente o time faz uma reunião de 15 minutos para acompanhar o progresso do trabalho e agendar outras reuniões necessárias. Na reunião diária, denominado Daily Scrum Meeting (DSM), cada membro do time respondia a três questões: O que eu fiz no projeto desde a última reunião? O que irei fazer até a próxima reunião? Quais são os impedimentos? Ao final da Sprint, é realizada a reunião de revisão (Sprint Review Meeting) para que o Time apresente os resultados alcançados na iteração ao Product Owner. Neste momento as funcionalidades são inspecionadas e adaptações do projeto poderiam ser realizadas. Em seguida, o ScrumMaster conduz a reunião de retrospectiva (Sprint Retrospective Meeting), com o objetivo de melhorar o processo/time e/ou produto para a próxima Sprint.

2.2.2 A Engenharia de Requisitos

Inicialmente conceituaremos o que vem a ser a engenharia de requisitos: podemos dizer que é uma disciplina para desenvolver uma especificação completa, consistente e não ambígua, que sirva como base para um acordo entre todas as partes envolvidas, descrevendo o que o produto de software irá fazer, e não como será feito. Então, requisitos de sistemas, definem o que o sistema é requerido a fazer e sob quais circunstâncias eles serão requeridos a operar [22].

Os requisitos são categorizados em dois grandes grupos: os requisitos funcionais e os requisitos não-funcionais.

- **Os requisitos funcionais** descrevem as funcionalidades ou serviços do sistema; possuem efeito localizado pois afetam somente a parte do sistema onde as funcionalidades são implementadas.
- **Os requisitos não funcionais** descrevem restrições ou atributos de qualidade relacionados ao sistema; fixam restrições sobre como serão implementados os requisitos funcionais e têm efeito sobre todo o sistema, pois a sua satisfação afeta vários componentes do sistema.

Os requisitos não funcionais se caracterizam pela subjetividade, isto é, podem ter significados distintos para pessoas distintas; relatividade, pois sua interpretação e importância dependem de cada sistema e sua realização é relativa e interatividade, pois interagem entre si, afetando negativamente ou positivamente uns aos outros, e podem ser subdivididos em:

- Requisitos de **usabilidade**, que estão relacionados a interface com o usuário, material de treinamento, documentação do sistema, fatores humanos, ajuda;
- Requisitos de **confiabilidade**, que estão ligados a frequência de falhas, capacidade de recuperação, previsibilidade, disponibilidade de uso, grau de tolerância a falhas, correteude do sistema;
- Requisitos de **desempenho**, relacionados com eficiência, tempo de resposta de processamento, uso de recursos computacionais, fluxo de vazão (throughput), precisão, disponibilidade, uso de recursos;
- Requisitos de **segurança**, relacionados com privacidade, integridade e autenticidade dos dados do sistema;
- Requisitos de **distribuição**, que se relacionam com a distribuição da versão executável do sistema. Requisitos de distribuição são críticos para sistemas com grande volume de usuários;
- Requisitos de **facilidade de suporte**, facilidade de **adaptação** e de **manutenção**, internacionalização, configurabilidade;
- Requisitos de **implementação**, limitações de recursos, linguagens e ferramentas, hardware, etc.;
- Requisitos de **interface**, restrições impostas pelas interfaces com sistemas externos;
- Requisitos de **operações**, que estão relacionados ao gerenciamento do sistema no ambiente operacional;
- Requisitos **legais**, que estão relacionados a licenças de uso.
- **Restrições** de hardware e software, que estão relacionados ao hardware e software para desenvolver e executar o sistema inclui a plataforma do cliente, a plataforma do servidor, protocolo de comunicação, etc.

Uma classificação para os requisitos não-funcionais seria em:

- **Requisitos de produto**, que são requisitos que especificam que o produto entregue tem que se comportar de um modo particular. Por exemplo, velocidade de execução, confiabilidade, usabilidade, etc.;
- **Requisitos de processo**, que são requisitos que estão relacionados com restrições sobre o processo de desenvolvimento do sistema (são uma consequência de políticas e procedimentos organizacionais). Por exemplo, padrões de processos usados, requisitos de implementação, entre outros;
- **Requisitos externos**, que são requisitos que surgem de fatores que são externos ao sistema e ao seu processo de desenvolvimento. Podemos citar como exemplo, exigências de interoperabilidade e requisitos legislativos.

Não é incomum os requisitos apresentarem alguns problemas, dentre os problemas podemos citar o fato dos requisitos não refletirem as reais necessidades dos clientes do sistema; dos requisitos serem inconsistentes ou incompletos; o alto custo para se fazer mudanças nos requisitos depois de terem sido concordados; existirem mal entendidos ou ambigüidades entre clientes, aqueles que desenvolvem os requisitos do sistema e os engenheiros de software que desenvolvem os requisitos do sistema e os engenheiros de software que desenvolvem ou mantêm o sistema, além de conflitos entre requisitos diferentes serem comuns em sistemas complexos [23].

2.2.2.1 O processo de Engenharia de Requisitos

O processo é organizado em atividades que se transformam em entradas e saídas. A descrição do processo é uma atividade importante, pois permite que o conhecimento seja reutilizado em outras situações [22], ou seja, uma vez que alguém tenha enfrentado um problema e essa solução tenha sido documentada, isso poderá ajudar outras pessoas que passem pelo mesmo tipo de problema em outro momento.

Podemos traduzir o processo em um diagrama de entradas e saídas que seria representado da seguinte forma:

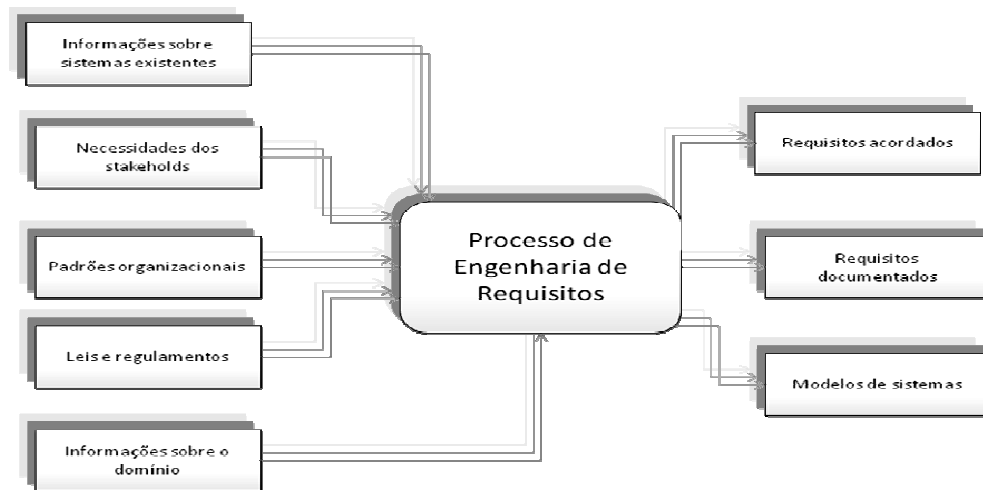


Figura 2- O processo da Engenharia de Requisitos: entradas e saídas. Fonte: SOMMERVILLE [22].

As entradas do processo são:

- **Informações sobre sistemas existentes:** são informações sobre as funcionalidades de sistemas que serão substituídos ou de sistemas que farão uma iteração com o sistema especificado;
- **Necessidade dos stakeholders:** Descrição de quais sistemas os *stakeholders* precisam para o desenvolvimento do sistema em questão;
- **Padrões organizacionais:** Padrões que são usados pela organização e que deveram ser levados em consideração durante o desenvolvimento da aplicação, gerencia de qualidade, metodologia etc.;
- **Leis e regulamentos:** Questões legais, que são externas ao sistema devem ser observadas, para que regras não sejam violadas;
- **Informações sobre o domínio:** informações gerais a respeito do domínio da aplicação.

Sobre as saídas do processo, temos:

- **Requisitos acordados:** Uma descrição dos requisitos do sistema que foram entendidos pelos *stakeholders* e acordados entre eles;
- **Requisitos documentados:** Isto será uma especificação mais detalhada do sistema a respeito do que pode ser produzido;
- **Modelos de sistemas:** é um conjunto de diagramas que descrevem o sistema de ângulos distintos.

Temos que considerar agora as 4 (quatro) **etapas do processo** de engenharia de requisitos, que são:

- Elicitação;
- Análise e negociação;
- Documentação;
- Validação.

Essas etapas não são desenvolvidas apenas uma vez cada no processo de desenvolvimento de software, e sim de forma iterativa e incremental (Figura 3), no modelo espiral.

- Na **elicitação de requisitos**, que é a primeira fase do processo, deve-se descobrir o máximo de informações para o conhecimento do objeto em questão, ou seja, é dever da fase de elicitação identificar os fatos relacionados aos requisitos do sistema, de forma a prover o mais correto e mais completo entendimento do que é demandado daquele software;
- Na segunda fase do processo, temos a **análise e negociação de requisitos**, que tem por objetivo descobrir problemas e inconsistências nos requisitos elicitados, além de descobrir as interações entre os requisitos e informar os conflitos e sobreposições encontrados. Vale ressaltar, que a análise será intercalada com a elicitação, pois problemas são descobertos quando os requisitos são elicitados;

- Na fase de **documentação dos requisitos**, teremos a produção do documento oficial que descreve os requisitos do sistema (Figura 4), esse documento define o que o sistema deve fazer e não como fazer, também serve como um acordo contratual entre os clientes e fornecedores de um software;
- Na última fase do processo, temos a **validação dos requisitos**, esta fase tem o objetivo de mostrar que os requisitos realmente definem o que os usuários realmente desejam do sistema, certifica que o documento de requisitos é uma descrição aceitável do sistema a ser implementado.

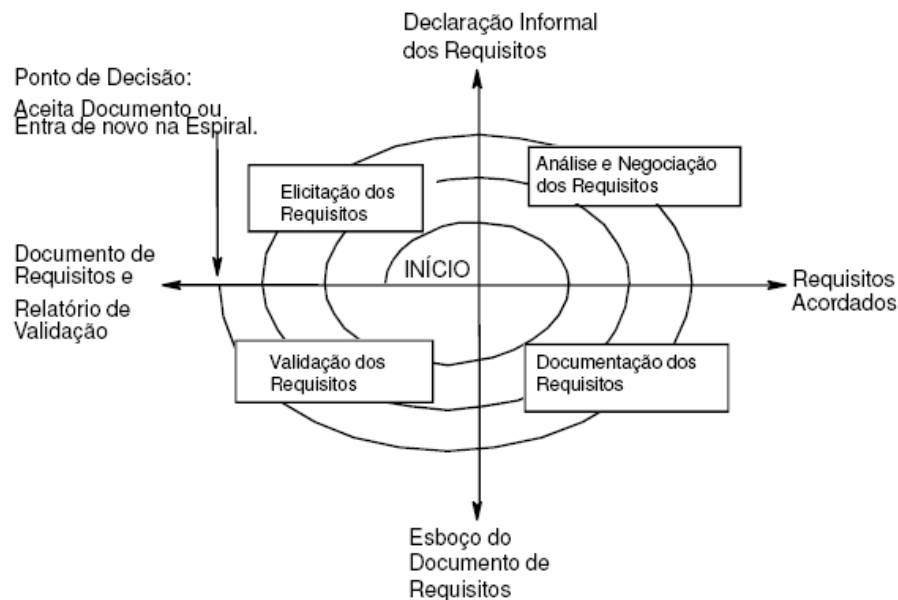


Figura 3- Modelo Espiral aplicado à Engenharia de Requisitos. Fonte: SOMMERVILLE [22].

Na primeira etapa do processo, que é a de **elicitação de requisitos**, destacamos algumas dificuldades que podem surgir como, por exemplo, alguns usuários podem não ter uma idéia precisa do sistema que eles desejam ou alguns usuários têm dificuldades para descrever seu conhecimento sobre o domínio do problema. Também deve ser considerado, o fato de usuários e analistas terem diferentes pontos de vista do problema, o que podem gerar uma elicitação diferente para o mesmo problema. Existe ainda outro fator complicador na fase de elicitação de requisitos, que é a possibilidade de algum usuário antipatizar-se com o novo sistema proposto, e se negar a participar da

elicitação ou o que seria pior, fornecer informações incorretas. Em softwares médicos, uma elicitação de requisito errada, se chegar a implementação do sistema, pode trazer sérios riscos para saúde humana. Dedicaremos uma seção posterior sobre as técnicas de elicitação de requisitos e a faremos com as devidas observações referentes a softwares médicos.

Com relação aos estágios do processo de elicitação, temos de início a definição dos objetivos, ou seja, os objetivos organizacionais devem ser estabelecidos incluindo objetivos gerais do negócio, uma descrição geral do problema a ser resolvido, o porquê o sistema é necessário e as limitações do sistema. Em seguida temos como segundo estágio a aquisição de conhecimento do contexto, isto é, essa fase inclui informações a respeito da organização onde o sistema será implantado, o domínio da aplicação do sistema e a informação acerca de outros sistemas existentes. Tem-se ainda o estágio da organização do conhecimento, onde a grande quantidade de conhecimento, que foi coletada nos estágios anteriores, deve ser organizada e ordenada, de forma que o conhecimento não se perca durante o desenvolvimento do projeto. Por fim tem-se o estágio de coleta dos requisitos dos *stakeholders*, onde os *stakeholders* do sistema são consultados para a descoberta dos seus requisitos.

Na etapa de **análise de requisitos**, temos um checklist que deve ser seguido, a primeira verificação é se o projeto é imaturo, ou seja, se os requisitos incluem informação prematura de projeto ou implementação, depois se faz a observação dos requisitos combinados, ou seja, verifica-se se a descrição do requisito descreve um único requisito ou se este pode ser descrito em vários diferentes. Em seguida uma análise crítica é feita para se chegar à conclusão de quais requisitos são realmente necessários e quais não são. Também se deve observar se os requisitos não implicam no uso de uma plataforma de hardware não padronizada. Deve-se ainda verificar se os requisitos estão consistentes com os objetivos de negócio definidos no documento de requisitos. Se existem requisitos que podem gerar interpretações distintas, isto é, se existem requisitos

ambíguos. O realismo dos requisitos deve ser checado, ou seja, o requisito é realístico em relação à tecnologia usada para a implementação do sistema e por fim se o requisito pode ser testado.

A **negociação de requisitos** é o processo de discussão dos conflitos de requisitos e a busca de um compromisso no qual todas as partes interessadas concordem, portanto, no planejamento do processo de requisitos, é importante deixar tempo para a negociação. Alcançar um compromisso aceitável pode tomar um grande tempo.

A negociação se divide em e (três) estágios:

1. **Discussão dos requisitos:** Os requisitos que foram identificados como problemáticos são discutidos e os *stakeholders* envolvidos apresentam seus pontos de vista a respeito desses requisitos;
2. **Priorização dos requisitos:** Os requisitos disputados são priorizados para identificar requisitos críticos e ajudar o processo de tomada de decisão;
3. **Concordância dos requisitos:** soluções para os problemas dos requisitos são identificadas e um conjunto de requisitos é acordado. Geralmente isto envolve mudanças em alguns dos requisitos.

Na terceira fase do processo, que é a fase da **documentação de requisitos**, destacamos como essencial na escrita de um documento de requisitos:

1. Requisitos são lidos mais freqüentemente do que são escritos. Portanto, deve-se investir tempo lendo e entendendo os requisitos no documento;
2. Não se deve assumir que todos os leitores dos requisitos tenham o mesmo conhecimento e usem a mesma terminologia que foi empregada pelo analista;
3. Deve-se permitir tempo para a revisão e ajustes no documento de requisitos.



Figura 4-Estrutura de um documento de requisitos. Fonte: IEEE [24]

Nesta fase, alguns problemas com a documentação podem ser observadas, como falta de clareza na escrita, é difícil ter a precisão sem tornar o documento difícil para leitura; confusão entre requisitos, requisitos funcionais e não funcionais tendem a ser misturados; fusão de requisitos, onde vários requisitos diferentes podem ser expressos juntos; ambigüidade, quando leitores e escritores do requisito devem interpretar as mesmas palavras, da mesma maneira; flexibilidade, o requisito deve está escrito no documento de uma forma que possa permitir que a mesma coisa possa ser dita de maneiras diferentes na especificação; por fim a falta de modularização, ou seja, estruturas de linguagem natural são inadequadas para estruturar requisitos do sistema.

Na quarta fase do processo que é a **validação dos requisitos**. Este processo trata, tal como o seu nome indica, da validação quanto à consistência, precisão, contextualização de requisitos levantados no processo de identificação e descoberta e de análise e negociação de requisitos. Este processo envolve uma revisão de todos os requisitos levantados e negociados, assim como uma prototipagem e validação de modelos e teste de requisitos.

Alexandre José Henrique de Oliveira Luna

Este processo é um dos mais importantes na Engenharia de Requisitos. Isto porque tal como um documento de requisitos bem definido permite a correção de incoerências e inconformidades no desenvolvimento de um produto de software, a validação permite minimizar o tempo gasto na detecção dessas incoerências e inconformidades devido à sua alta eficiência na sua descoberta. Também porque como é este processo que permite a identificação destas mesmas incoerências na fase anterior à versão final do relatório de requisitos, minimiza grandemente o risco de encontrar estas incoerências numa fase tardia, ou até mesmo na terminação, do desenvolvimento do sistema. É fácil entender que um erro encontrado numa fase tardia do desenvolvimento do projeto pode ser desastroso, pois a sua alteração poderá ser bastante onerosa, se não intolerável, em termos temporais.

Poderemos afirmar que o processo de validação de requisitos está para o documento de requisitos assim como a fase de testes unitários e de sistema está para a fase de desenvolvimento de um projeto de software.

Discutiremos essas fases dando ênfase a softwares aplicados a telemedicina/telessaúde, mais à frente.

Capítulo 3

3 Engenharia de Requisitos aplicada à Saúde

Como já citado anteriormente, cabe a **elicitação** a tarefa de identificar os fatos relacionados aos requisitos do sistema, de forma a prover o mais correto e mais completo entendimento do que é demandado naquele software. Para conseguir chegar ao que foi dito, podemos escolher ou combinar algumas das técnicas que serão apresentadas, de acordo com o tipo de software e o tipo de cliente envolvido na construção do produto. É importante lembrar, que a fase elicitação de requisitos não se limita a apenas perguntar as pessoas o que elas querem, e sim uma análise cuidadosa da organização, observando o domínio da aplicação e do processo do negócio em que o sistema será usado [22].

O termo elicitação de requisitos sugere que o processo se resume a simples transferência de conhecimento das pessoas para os documentos. Onde na realidade o processo é muito mais complexo [22]. Os clientes raramente têm uma idéia clara dos requisitos e alguns requisitos propostos freqüentemente possuem limitações tecnológicas. Portanto, a elicitação de requisitos não é simplesmente um processo de “pescar” os requisitos. E sim um complexo processo de negociação envolvendo todos os *stakeholders*, incluindo os desenvolvedores de sistemas [30].

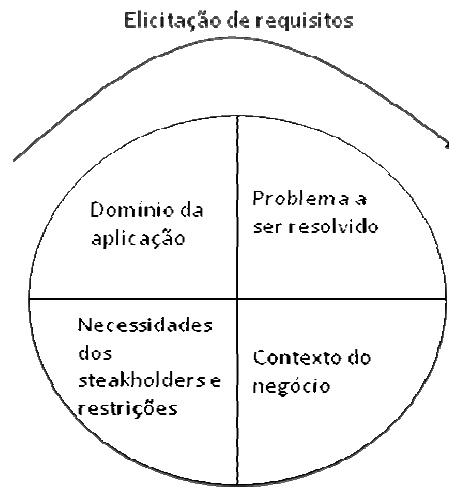


Figura 5 - Processo de elicitação de requisitos. Fonte: SOMMERVILLE [22].

O processo de elicitação de requisitos, pode ser subdividido em 4 (quatro) etapas, conforme a **Figura 5**. São elas:

1. **Domínio da aplicação:** É o conhecimento geral da área onde o sistema será aplicado. No caso de softwares médicos, que serão utilizados em um hospital, por exemplo, deve-se ter um conhecimento mínimo sobre o funcionamento de um hospital, bem como conhecimento dos termos usados no dia-a-dia de uma instituição de saúde;
2. **Problema a ser resolvido:** Detalhes dos problemas dos clientes onde o sistema será aplicado devem ser entendidos [30]. Para um software para prontuário eletrônico de paciente, por exemplo, deve-se aprender com o cliente as dificuldades encontradas para se preencher atualmente essas informações;
3. **Contexto do negócio:** Sistemas pretendem contribuir de alguma maneira no desenvolvimento do negócio ou da organização. Portanto deve-se entender como esses sistemas interagem e afetam as diferentes partes do negócio e como pode contribuir para o objetivo global da organização [30];
4. **Necessidades dos stakeholders:** Os *stakeholders* do sistema são aquelas pessoas que são afetadas de alguma maneira pelo sistema. Podem ser

Alexandre José Henrique de Oliveira Luna

usuários finais do sistema, gerentes de departamentos onde o sistema será implantado etc. Devem-se entender, em detalhes, as necessidades específicas para que o sistema suporte-as no seu trabalho. Em particular, o engenheiro de requisitos deve entender o processo que o sistema pretende suportar e as regras de sistemas existentes nesse processo de trabalho.

A correta **elicitação de requisitos** é essencial. Se “as necessidades” dos clientes não forem descobertas, entendidas, estruturadas e validadas, o sistema final não irá atender com plenitude o usuário final. A aceitabilidade do sistema depende do quanto “precisas” forem identificadas e desenvolvidas as necessidades dos clientes.

3.1 Técnicas para Elicitação de Requisitos

Estaremos apresentando e discutindo algumas técnicas de elicitação de requisitos, e em cada uma das técnicas faremos uma análise crítica em relação a sua eficácia para o caso de softwares aplicados a saúde, onde temos clientes tão especiais, como médicos, enfermeiros, odontólogos ou gestores de saúde – secretários municipais, estaduais, diretores de postos de saúde etc.

a. Entrevistas e questionários: De acordo com Sommerville [22], entrevista é a mais comum das técnicas usadas na elicitação de requisitos. O analista discute o sistema com diferentes *stakeholders* e constroem juntos o entendimento dos requisitos. Existem basicamente dois tipos de entrevistas:

1. As **entrevistas fechadas**, onde o analista de requisitos já tem uma lista pré-definida de perguntas;

2. E as **entrevistas abertas**, onde não existe um conjunto pré-definido de perguntas, e as perguntas são feitas de acordo com a vontade dos *stakeholders*.

Para softwares médicos, a técnica de entrevista se mostra bastante eficiente, porém deve-se ter cuidado extra com os termos médicos e as ambigüidades nas palavras do entrevistado. Isso só torna mais relevante a necessidade do analista ter um conhecimento prévio do negócio e da organização para a qual o sistema está sendo desenvolvido e será implantado.

- b. Brainstorm:** Essa técnica poderia também ser chamada de tempestade de idéias, e consiste em reunir em uma sala stakeholders do processo e acolher todas as idéias que surjam durante o processo, para só depois o analista filtrar o que for pertinente e o que não for. Nessas reuniões não é indicada a participação de muita gente. Devem-se incluir pessoas com diferentes perfis e a presença de um facilitador. O analista não deve pensar em detalhes durante a reunião, todos os presentes devem ser perguntados e sugestões podem ser dadas pelo analista.
- c. Cenários:** Usuários finais e demais stakeholders acham simples relatar o trabalho cotidiano e assim descrever as funcionalidades que deverão ser providas pelo sistema [22]. Em outras palavras, cenários são histórias que explicam como um sistema poderá ser usado. Quando um sistema é muito grande, o número de cenários também é muito grande e complexo. Cenários podem ser escrito de várias maneiras, mas devem incluir, pelo menos, as seguintes informações:
 1. Uma descrição do estado do sistema antes de entrar no cenário;
 2. O fluxo normal de eventos no cenário;

3. Exceções do fluxo normal de eventos;
4. Informações sobre outras atividades que podem ocorrer no mesmo momento;
5. Uma descrição do estado do sistema após completar o cenário.

A descrição de cenários é também interessante nos softwares aplicados a saúde, desde que o analista esteja bem apropriado do negócio em questão para que possa compreender o cenário descrito, por exemplo, do funcionamento do hospital ou do consultório, caso contrário, e que o cliente entenda bem o que deseja e que saiba se expressar de maneira correta, visto que nem todas as pessoas acham simples descrever em detalhes o que fazem.

d. Etnografia: Essa técnica provém das ciências sociais, e se mostrou útil no entendimento dos processos reais realizados nos trabalho. O etnógrafo passa algum tempo observando as pessoas no trabalho e constrói uma imagem de como o trabalho é realizado. Portanto o etnógrafo deve ser uma pessoa com boa habilidade de observação, tradução dessa observação em cenários ou até mesmo casos de uso, comunicação e sintetização de idéias. Podemos considerar algumas diretrizes para a etnografia, dentre elas destacamos:

1. Devem-se procurar formas não padronizadas de trabalho;
2. Algum tempo conhecendo e ganhando a confiança das pessoas deve ser dispensado;
3. Deve-se tomar nota de forma detalhada, de todas as práticas de trabalho, analisá-las e chegar a uma conclusão;
4. Recomenda-se combinar etnografia com entrevistas abertas e outras técnicas de elicitção.

No caso de softwares aplicados a saúde, em alguns tipos de sistemas a etnografia se torna difícil para o analista, visto a especificidade do negócio e a dificuldade de acompanhamento das atividades de um médico, por exemplo.

- e. Reuso de requisitos:** considerando que é uma boa prática da engenharia de software o reuso como aproveitamento do conhecimento gerado. O reuso de requisitos também o é. O reuso de requisitos economiza tempo e esforço, pois requisitos reutilizados já foram analisados e validados em outros sistemas.

- f. Prototipação:** O protótipo de um sistema e a versão inicial do sistema que será avaliado primeiro durante o processo de desenvolvimento. Em sistemas de software, protótipos são freqüentemente usados para ajudar na elicitação e validação dos requisitos do sistema. Outro fator a ser considerado, é o desenvolvimento rápido dos protótipos, ou seja, é fundamental que disponíveis logo para o processo de elicitação. Como benefícios da prototipação, podemos destacar:
 1. O protótipo permite que os usuários experimentem e descubram o que eles realmente necessitam para suportar o trabalho deles;
 2. Estabelece a viabilidade e utilidade antes que altos custos de desenvolvimento tenham sido realizados;
 3. Essencial para desenvolvimento de uma interface de usuário que esteja de acordo com as necessidades dos usuários do sistema;
 4. Pode ser usado para teste do sistema e desenvolvimento da documentação;
 5. Força um estudo detalhado dos requisitos, revelando inconsistências e omissões.

Alexandre José Henrique de Oliveira Luna

A prototipação tem se mostrado uma técnica bastante eficiente para a elicitação de requisitos nos softwares aplicados a saúde [4, 5, 33, 34].

Geralmente o profissional da área de saúde, mesmo sendo o fornecedor das informações para elaboração do produto, não se sente confortável com a documentação dos requisitos e muitas vezes acabam se confundindo e causando retrabalhos – o que até certo ponto é esperado, visto que requisitos mudam. Através de protótipos podemos simular o funcionamento real do sistema, e isso servirá de base para que o cliente se sinta bem mais confortável para criticar, sugerir mudanças, correções e a descoberta da necessidade de mais requisitos, sendo a descoberta de novos requisitos, crucial para o sistema em desenvolvimento visto que antes disso o sistema estaria incompleto.

O protótipo assim como o software a ser desenvolvido, deve estar em constante evolução e nunca ser entregue ao cliente no final do processo, ou seja, no início apenas alguns requisitos serão elicitados e validados, mas posteriormente o novo protótipo contará com novas interfaces para que o usuário final possa observar se está de acordo ou não e se novos requisitos não são pertinentes [4].

Porém, existem alguns custos e problemas associados à prototipação, e esses problemas devem ser analisados sob o ângulo do quanto vale à pena, mesmo com o custo, se investir nessa técnica. Dentre os problemas, destacamos:

- **Custo do treinamento:** Se a empresa não possuir experiência no desenvolvimento de protótipos, os desenvolvedores devem ser treinados para usar um ambiente de desenvolvimento de protótipos ou ser treinado em uma das técnicas de prototipação;

- **Custo de desenvolvimento:** Este custo está associado à necessidade de realocar alguns desenvolvedores da tarefa de desenvolvimento para a produção do protótipo, ou dependendo do tamanho da aplicação e do protótipo, a equipe pode ser ainda maior;
- **Extensão do cronograma:** Em alguns casos, o desenvolvimento de um protótipo causa o alongamento do prazo de entrega do produto;

Alguns clientes pensam que o sistema funcionará exatamente igual ao protótipo, com a mesma velocidade, estabilidade etc. Mas sabemos que isso não se reflete na realidade, o que chega a frustrar o cliente em alguns casos. Portanto deve-se deixar bem claro pra o cliente, que o protótipo não é nada que ele vai poder comparar em relação às interfaces [4].

3.2 Técnicas de Modelagem de Requisitos

De acordo com Pressman [28], modelos são representações tanto do mundo físico como do mundo lógico, são feitos usando-se diagramas que representam graficamente o domínio, melhorando o entendimento das informações adquiridas, permitindo a avaliação dos problemas atuais e das informações desejadas como entrada/saída. A origem destes problemas é que os requisitos são a interface entre o cliente e o analista. Para tal, é necessário o uso de representações abstratas que descrevam o funcionamento e a interação do sistema. Através destas representações abstratas é criado um modelo do mundo real, ou da parte dele correspondente ao sistema, sendo que os requisitos devem ser capazes de descrever sistemas grandes e complexos.

De acordo com Kondo [29], podemos destacar a seguir as técnicas de modelagem mais expressivas, baseadas em diferentes tipos de abstração:

- a. **B [38]**: a base para a modelagem de um sistema em B é a máquina abstrata. Em outras palavras, a modelagem de um sistema é uma associação lógica de máquinas abstratas onde: vale o princípio da composicionalidade, isto é, uma associação de máquinas abstratas é também uma máquina abstrata;
- b. **Casos de Uso**: uma descrição de um conjunto de seqüências de ações, resultantes da interação do sistema com um ator (um tipo de requerente). As ações produzem um resultado que pode ser observado pelo ator;
- c. **CORE – Controlled Requirements Expression**: domínio a ser modelado é limitado e particionado em perspectivas, que representam organizações, homens, entidades de hardware ou software;
- d. **DFD – Diagrama de Fluxo de Dados**: concentra-se no fluxo de dados e nas transformações funcionais dos dados, deixando de lado detalhes da estrutura de dados;
- e. **Diagrama de Contexto**: representação do sistema por um único processo e suas iterações com entidades externas;
- f. **ERAE – Entidade-Relação-Atributo-Evento, do inglês Entity-Relation-Attribute-Event**: direciona a definição dos requisitos do prisma “orientado-a-computador” para “orientado-a-problema”. Como resultado: a validação e a completude são facilitadas;
- g. **KBRA – Assistente de Conhecimento Baseado em Requisitos, do inglês Knowledge-Based Requirements Assistant**: subsistema do knowledge-based software assistant, nascido da aspiração de criar um paradigma de ciclo-de-vida baseado em conhecimento, capaz de formalizar os requisitos usados no desenvolvimento;
- h. **MAL -Lógica da Ação Modal, do inglês Modal Action Logic**: definido como uma tentativa de produzir o formalismo matemático necessário à especificação de requisitos, sem que para tal fosse necessário o conhecimento matemático para escrever e entender as especificações;

- i. **Modelo de Estímulo-Resposta:** diagramas de transição de estado mostram como o sistema reage a eventos internos e externos;
- j. **Modelo do Processo:** mostra as principais atividades do processo;
- k. **QFD – Quality Function Deployment** [39]: um método de avaliação de qualidade dos produtos que precedesse ao processo de manufatura e mesmo ao design;
- l. **Redes de Petri – Petri Nets** [40]: formadas por dois tipos de componentes: a transição, componente ativo correspondente a alguma ação realizada dentro do sistema, e o lugar, passivo e relacionado a alguma variável de estado do sistema;
- m. **RLP – Requirements Language Processor:** na tentativa de solucionar o problema dos diversos tipos de leitores, o RLP permite o desenvolvimento de linguagens formais para os requisitos de especificação das aplicações;
- n. **RML – Requirements Modelling Language:** Assim como a ERAE, procura dar destaque ao ambiente do sistema. Sua especificação consiste numa série de objetos relacionados, que procuram representar conceitos do mundo real. Os objetos são agrupados em classes, que agrupam características comuns;
- o. **SCS – Structured Common Sense:** foi desenvolvido como um método de auxílio ao desenvolvimento de especificações formais a partir de conceitos da aplicação. É dirigido para extração e formalização dos requisitos usando o MAL;
- p. **UML – Unified Modeling Language** [25]: linguagem para visualizar, especificar, construir e documentar os requisitos e informações de um sistema;
- q. **Pontos de Vista, do inglês Viewpoints** [28]: mecanismo que permite considerar aspectos do sistema percebidos por diferentes requerentes. Ao se analisar o sistema por vários aspectos, obtém-se uma especificação mais

adequada, a qual permite um melhor entendimento das necessidades dos usuários através de cada uma das atividades do processo;

- r. **Volere** [41]: é um método completo de obtenção de requisitos, baseado nos casos de uso.

3.3 Técnicas de Validação de Requisitos

Como já abordamos anteriormente, a **validação de requisitos** tem o objetivo de mostrar que os requisitos realmente definem o que os usuários desejam do sistema, ou seja, certifica que o documento de requisitos é uma descrição aceitável do sistema a ser implementado. A validação de requisitos é a etapa final da engenharia de requisitos [22]. Este processo envolve os *stakeholders*, engenheiro de requisitos e design de sistemas que: analisaram os problemas, omissões e ambigüidades dos requisitos. Após a análise dos requisitos, teremos como elementos de saída:

1. **Lista de problemas:** É uma lista onde estarão documentados os problemas encontrados nos requisitos, e se possível os erros deverão ser classificados quanto ao tipo, por exemplo: ambigüidade, incompleto, desnecessário. Entretanto, freqüentemente essa classificação dos problemas é difícil de ser feita;
2. **Lista de ações:** É uma lista onde estarão documentadas as ações que serão tomadas para que a lista de problemas seja resolvida, e quem será o responsável por cada resolução do problema.

Agora, descreveremos algumas **técnicas de validação de requisitos** usadas no desenvolvimento de sistemas e sempre que possível faremos uma análise dessa técnica junto a clientes de softwares aplicados a saúde.

- a. **Prototipação:** Assim como pode ser usada para a elicitação de requisitos, protótipos podem ser usados para a validação de requisitos, neste caso um protótipo do sistema é mostrado aos usuários finais e clientes. Eles podem experimentar o protótipo e verificar se ele atende às suas necessidades. O protótipo para validação deve ser completo, eficiente e robusto e deverá ser possível usá-lo da mesma forma que o sistema requerido.
 - i. Esta técnica tem se mostrado a mais eficiente para os softwares aplicados a saúde, pois consegue mostrar como o requisito funcionará quando implementado, o que deixa o cliente bem mais confortável do que documentos com diagramas ou textos, principalmente profissionais da área médica que na maioria das vezes são avessos a softwares [4, 5];

- b. **Validação de modelos:** Nesta técnica, os requisitos são documentados utilizando linguagem natural e diagramas. Todos esses documentos precisam ser validados entre si para garantir a consistência e a correteude em relação às necessidades dos stakeholders.
 - i. Essa técnica não funciona muito bem com profissionais da área de saúde, na maioria das vezes tanto o documento textual quando os diagramas não são entendidos, ou freqüentemente são entendidos de forma errada, o que gera problemas posteriores, quando a funcionalidade já está implementada [4, 5].

- c. **Geração de casos de teste:** Casos de teste são projetados a partir dos requisitos. Se um teste é difícil ou impossível de ser projetado, isso pode significar que o requisito correspondente é de difícil implementação e deve ser reconsiderado.
- i. O usuário do sistema pode sugerir casos de teste para a aplicação, mas não é comum que médicos, enfermeiros ou odontólogos tenham condições de sugerir casos de teste para testar seus requisitos, portanto torna-se uma técnica muito boa para a equipe de desenvolvimento, mas não para os usuários finais dos produtos [4, 5].

3.3.1 Revisão de Requisitos

Temos ainda uma atividade a ser considerada, que é a de **revisão de requisitos**, onde um grupo de pessoas lê e analisa os requisitos, procura problemas, se reúne, discute os problemas e chegam a um acordo nas ações para tratar estes problemas. Os revisores devem incluir *stakeholders* com perfis diferentes e o time de revisão deve sempre incluir um especialista no domínio e um usuário final.

Como atividades da revisão, temos:

- a. **Planejar a revisão:** selecionar o time de revisão, hora e local para que o encontro seja realizado;
- b. **Distribuir documentos:** os documentos a serem revisados são distribuídos entre os membros do time de revisão;
- c. **Preparação para revisão:** Cada revisor individualmente deve ler os requisitos e encontrar conflitos, omissões, inconsistências e desvios dos padrões e outros problemas;
- d. **Realizar encontro de revisão:** Os problemas e comentários individuais são discutidos e um conjunto de ações para tratar dos problemas é acordado;

- e. **Ações de acompanhamento:** O líder da revisão deve verificar se todas as ações acordadas estão sendo executadas;
- f. **Executar nova revisão dos documentos:** os documentos são atualizados para refletir as ações concordadas e em seguida revisados novamente. Neste estágio, podem ser aceitos ou revisados novamente. Esta revisão pode ser feita por um revisor indicado no encontro da revisão.

A validação de requisitos é uma etapa longa do processo e envolve várias pessoas que possam ler e pensar a respeito dos documentos. Existe sempre a tendência natural de querer apressar o processo, principalmente quando o cronograma está apertado. Entretanto, se o sistema em desenvolvimento começou muito cedo, algum retrabalho será quase inevitável quando os problemas com os requisitos surgirem [22].

Vale ressaltar que no caso específico de softwares médicos, um requisito elicitado, validado e implementado de forma incorreta, pode custar a vida de pessoas, daí a grande importância dessas etapas do processo de desenvolvimento de softwares aplicados a saúde [4, 5, 29].

Capítulo 4

4 Particularidades da abordagem da Engenharia de Requisitos para softwares aplicados à Saúde

A indústria de sistemas de telemedicina vem crescendo e expondo seus produtos como essenciais para uma prestação de serviços de saúde mais eficientes e de melhor qualidade. No entanto, apesar deste crescimento, o nível de utilização desses sistemas ainda está longe de atingir patamares elevados [43], particularmente na saúde pública brasileira.

As questões que se colocam são: quais os requisitos para que sistemas de telemedicina tenham um alto índice de utilização? Qual a qualidade da informação e sua forma de transmissão? A interface é amigável? É de fácil utilização? Tem confiabilidade? Ou, principalmente, qual a sua aderência às necessidades do cliente/usuário?

Todas estas questões e muitas outras poderiam, e deveriam ser colocadas durante o planejamento do desenvolvimento dos softwares de telemedicina. No entanto a abordagem de alguns destes fatores só tem se demonstrado viável, apenas quando estes softwares são disponibilizados para os usuários. Em função desta verificação as técnicas de prototipagem ganham força no processo de desenvolvimento de softwares para telemedicina e telessaúde.

4.1 Aspectos que influenciam a aplicação de Engenharia de Requisitos no domínio da Saúde

Abaixo estão elencados os diversos aspectos envolvidos e uma breve análise em torno de cada um deles [4]. Partindo das pessoas envolvidas nesses tipos de projeto,

avaliando os fatores relevantes, muitas vezes extremamente conflitantes, entre suas formações e expectativas.

Na seqüência encontraremos uma discussão sobre a engenharia de requisitos, tendo por foco como ela é vista sob o ângulo de profissional da área de Tecnologia da Informação e Comunicação - TIC e de um profissional de saúde, ou seja, um profissional de saúde, como cliente. Por fim faremos uma discussão sobre o uso e usuário do sistema, suas particularidades, expectativas a respeito do software.

4.1.1 O lado humano (Stakeholders)

No desenvolvimento de softwares, as pessoas envolvidas geralmente possuem formações distintas e diversas. Para este tipo de Software, ocorre com muito mais freqüência a existência de Equipes multidisciplinares, que são de natureza mais heterogênea que em projetos de outra área de aplicação.

Desta forma, podemos exemplificar os diversos tipos de profissionais envolvidos, organizados nas seguintes classes de stakeholders:

- Profissionais de TIC, de um modo geral:
 - Engenheiros de Sistema,
 - Engenheiros de Software,
 - Analistas de Sistemas,
 - Gerentes de Projeto,
 - Cientistas da Computação,
- Profissionais de Saúde, numa visão bem ampla:
 - Médicos Clínicos,
 - Médicos Especialistas (em suas diversas especializações),
 - Médicos Cirurgiões*,
 - Médicos Anestesiologistas,

- Enfermeiros,
- Psicólogos,
- Assistentes Sociais,
- Psicoterapeutas,
- Biomédicos,
- Psicanalistas,
- Fonoaudiólogos,
- Fisioterapeutas,
- Farmacêuticos,
- Nutricionistas, etc.
- Profissionais Administrativos, de uma forma bastante sintética:
 - Auxiliares Administrativos,
 - Operadores de equipamentos médicos;
 - Gestores de Saúde: estamos nos referindo aos profissionais que desempenham funções de gestão, independente da formação, como: Chefes de Unidades ou Equipes de Saúde, Diretores de Instituições de Saúde, Políticos no papel de Secretários de Estado ou Município relacionados à Saúde, etc.

De acordo com Norman [44], modelo mental é o nosso modelo conceitual particular da maneira como um objeto funciona, eventos acontecem ou pessoas se comportam; que resulta da nossa tendência de dar explicações para as coisas. Esses modelos são essenciais para nos ajudar a entender nossas experiências, prever reações de nossas ações e manipular ocorrências inesperadas. Nós baseamos nossos modelos no conhecimento que temos, real ou imaginário, ingênuo ou sofisticado.

Os profissionais de saúde, oriundos das Ciências da Saúde, muito próximas às Ciências Humanas, são formados através de uma escola filosófica, que incentiva o

Alexandre José Henrique de Oliveira Luna

desenvolvimento de certas características humanas onde a objetividade é fator secundário, como: relações humanas, o ouvir, investigar por anamnese, dentre outros.

Se partirmos da análise do processo de anamnese, teremos uma noção mais precisa da peculiaridade do modelo mental, no qual são forjados estes profissionais.

A anamnese (do grego *ana* = trazer de novo e *mnesis* = memória) é a parte mais importante da medicina pois envolve o núcleo da relação médico-paciente, onde se apóia a parte principal do trabalho médico; além disso preserva o lado humano da medicina e orienta de forma correta o plano diagnóstico e terapêutico. A anamnese, em síntese, é uma entrevista que tem por objetivo trazer de volta à mente todos os fatos relativos ao doente e à doença. Não é, no entanto, o simples registro de uma conversa. É mais que isto: é o resultado de uma conversação com um objetivo explícito, conduzido pelo médico e cujo conteúdo foi elaborado criticamente por ele. É a parte mais difícil do exame clínico. Seu aprendizado é lento, só conseguido após a realização de dezenas de entrevistas criticamente avaliadas. A anamnese é, na maioria dos pacientes, o fator isolado mais importante para se chegar ao diagnóstico.

Por outro lado os profissionais de saúde, especialmente médicos, são pessoas, com muita frequência, resistentes à adoção de um software para ajudá-los nas tarefas diárias, basta visitar alguns consultórios para constatar que a grande maioria dos médicos mantém o cadastro dos seus pacientes e o histórico de visitas anotado em papel, fichas, etc.

Apesar de *este aspecto* estar em franca melhoria sob a ótica da nova geração de profissionais de saúde em formação; de aspectos de colaboração da vida

Alexandre José Henrique de Oliveira Luna

cotidiana já abordar os conceitos da internet no nosso dia-a-dia; e de possuímos, hoje, bons profissionais da área de saúde engajados em pesquisas ou em projetos de desenvolvimento de softwares para aplicação na área de telessaúde e telemedicina; de uma maneira geral é “cultura” para os profissionais de saúde que o uso de recursos de informática, incluindo-se aí: dispositivos móveis, softwares específicos, uso de e-mail, navegação na Web e uso de sistemas de informação na área médica, ser um **tabu**.

Sob esta ótica os **cirurgiões** merecem um destaque especial entre os profissionais médicos: o de serem extremamente precisos no repasse de informações num processo de elicitación, pontuais em seus compromissos, exigentes nos resultados aguardados, mas “inacessíveis” na mesma intensidade.

O autor chegou a vivenciar, em sua experiência profissional, situações reais onde médicos forneceram suas senhas de acesso à auxiliares administrativos, por julgarem o processo de interação com os sistemas de informação, mero trabalho de digitação, quando na realidade estavam abrindo precedentes para serem responsabilizados por crimes de diversas naturezas, no caso de atitudes de má fé de seus colaboradores, que vão desde responsabilidade administrativa, lesa-economia na autorização indevida de procedimentos de alto custo e complexidade, quebra da preservação da confidencialidade dos dados do paciente, dentre outros.

Por outro lado, os Profissionais de TIC (informática) são oriundos das Ciências Exatas, onde sua formação ocorre através de uma escola filosófica que presa pela objetividade e aplicação suprema do raciocínio lógico, muitas vezes não possuindo muita capacidade de relacionamento com seus pares, ou aptidões sociais: que significa lidar bem com as emoções nos relacionamentos; interagir com facilidade; utilizar essas habilidades para liderar, solucionar conflitos, estabelecer vínculos de

cooperação e trabalho em equipe. Saber conviver com as pessoas de forma tranqüila, vendo a vida de forma proativa.

Costuma se empregar o termo “cartesiano” para os profissionais da área de Ciências Exatas com frequência, em referência ao filósofo René Descartes, por vezes chamado de "o fundador da filosofia moderna" e o "pai da matemática moderna", que é considerado um dos pensadores mais importantes e influentes da História do Pensamento Ocidental.

Estes profissionais passam a possuir a necessidade de serem flexíveis e maleáveis, pois encarando o paradoxo da informática como uma disciplina meio e não fim, esse profissional sempre precisará estar em contato com profissionais de outras áreas do conhecimento.

Enfim, observamos choques e complementos de dois modelos mentais, culturas e mundos, que precisam reconhecer e compreender suas diferenças, para poder colaborar de forma simbiótica e harmônica para construção de produtos de software que possam atingir a excelência na área de telemedicina e telessaúde.

4.1.2 Vários pontos de vista: um mesmo problema

Refletindo a respeito, podemos concluir que a primeira barreira a ser vencida é convencer o médico de que um software é um produto confiável e que vai melhorar a sua rotina de trabalho.

Superado este obstáculo, os próximos passos serão a definição do problema exato a ser resolvido pelo software a ser desenvolvido e o início do processo de elicitação dos requisitos.

Nesta fase iniciamos outra atividade crítica, pois os profissionais de saúde, muitas vezes, têm muita dificuldade em externar o que realmente precisam. Em função disso várias técnicas podem ser usadas, para tentar ajudar o cliente a estabelecer com clareza as suas necessidades.

Observamos que não são raras às vezes em que um requisito que foi discutido e revisado amplamente, pelos *stakeholders*, depois de implementado não é útil para cliente, ou seja, durante a comunicação entre analista e cliente não houve de fato um entendimento correto sobre o requisito.

Problemas assim geram retrabalho e encarecem o processo de desenvolvimento do software, então mesmo com todas as técnicas usadas, documentação pronta, contratos assinados, supostas visões convergentes, descobre-se, geralmente no final dos projetos de software, que *stakeholders* e a Equipe de TIC não falaram a mesma língua e que foi implementada alguma funcionalidade que não está de acordo com a real necessidade do cliente.

Um dos motivos para a ocorrência deste problema está no fato do desconhecimento do ciclo de vida de um projeto de desenvolvimento de software por parte do cliente. Cabendo ao profissional de TIC a responsabilidade de “educar” o seu cliente sobre como este processo acontece, durante o decorrer de seu ciclo. O que requer muita habilidade e talento.

Nestes termos podemos dizer que para o profissional de TIC, o processo de engenharia de requisitos é bastante claro e natural, o mesmo não se aplicando para o profissional de saúde, principalmente para aquelas etapas do processo onde o cliente precisa estar diretamente envolvido, como: elicitação, negociação e validação.

O profissional de saúde considera, na maioria dos casos estes processos extremamente nebulosos e complexos, chegando a questionar, em casos extremos, a necessidade de sua aplicação no projeto.

Para alguns profissionais de saúde, o fato do analista de sistemas chegar até ele e perguntar *o que ele deseja*, é similar ao fato um médico perguntar ao paciente qual a medicação que ele deseja que lhe seja prescrita. Em outras palavras é como se o profissional de TIC precisasse possuir toda uma bagagem de conhecimento anterior sobre o negócio, os processos, os procedimentos relacionados com a área de saúde abordada pelo projeto de software em questão. É claro que um profissional de TIC que se aperfeiçoa nesta área tende a alcançar este estágio de excelência, mesmo assim, as perguntas serão apenas mais sutis e direcionadas, e não deixarão de existir ao procurar se identificar quais requisitos precisam ser atendidos na construção de um determinado software.

Enfim, gostaríamos de chamar atenção para a influência destes pontos de vista distintos, sob um mesmo problema, e de como este aspecto pode influenciar e dificultar o processo de engenharia de requisitos para um projeto de desenvolvimento de software para aplicações no domínio da telessaúde e telemedicina.

4.1.3 O grande encontro: homem e software

Voltamos, nesta seção, a um aspecto já levantado anteriormente no início deste capítulo, quando nos referíamos ao fato de que alguns fatores da abordagem da Engenharia de Requisitos são viáveis, apenas, quando da disponibilização do software para os usuários.

Outros fatores críticos a série considerados, após a finalização do ciclo de desenvolvimento, são **os usuários** e **o uso do sistema**. Os usuários, suas visões de mundo, modelos mentais e as influências destes aspectos nos processos de Engenharia de Requisitos e no ciclo de desenvolvimento de software, já foram abordadas amplamente, anteriormente. Reforçamos aqui que, a depender do tipo de software em questão, os usuários nem sempre são da área médica, podem ser administradores, gestores, operadores de máquina médicas, etc.

Neste momento, quando o software é disponibilizado para o usuário, ocorre o encontro dos dois mundos, e as expectativas de ambas as partes são colocadas à prova. Uma abordagem segura é procurar manter o usuário envolvido deste o primeiro momento na visão concreta, embora parcial, do que está sendo construído para ele usar, realizando “entregas” parciais e freqüentes dos elementos do software que farão parte do dia-a-dia da vida dos stakeholders do projeto. Agindo assim, é possível realizar ajustes de conduta e redirecionamentos no projeto que sejam de pequeno impacto, por serem pontuais, na relação com os stakeholders, evitando que todo este conjunto de sentimentos, atitudes, intenções, expectativas e temores, fiquem concentrados num momento único de entrega do produto final. Minimizando assim o risco de insucesso do projeto.

Ou seja, caso o produto final, não esteja de acordo com o que foi solicitado ou não esteja tão atrativo, nem funcione tão bem quanto o esperado, estará fadado a entrar em desuso muito rápido. Neste ponto do processo o software recém-entregue será comparado imediatamente ao cenário antes da sua implantação. Se o resultado desta comparação não estiver dentro das suas expectativas, os usuários se tornarão cada vez mais resistentes para usar o sistema desenvolvido, até que o software cairá em desuso.

Alexandre José Henrique de Oliveira Luna

Para minimizar estes tipos de choque de expectativas, é papel da Engenharia de Requisitos garantir que o software estará exatamente como descrito e acordado no documento de requisitos, com os stakeholders. Na etapa de transição do Projeto, um treinamento deverá ser fornecido aos usuários do sistema e este deverá estar funcionando assim como especificado, na ocasião, sem que dados sejam perdidos ou violados, e sem que o sistema produza resultados inesperados.

Com a visão deste cenário nada simples, observamos a dificuldade em se produzir um software para aplicações dessa natureza. Onde um dos maiores desafios, neste contexto é implantar um sistema que seja exatamente o que o profissional deseja – quando muitas vezes, o cliente não consegue manifestar de forma clara, o que realmente deseja ou precisa - e que este software consiga de alguma forma melhorar o cotidiano de trabalho deste profissional de saúde.

A avaliação dos sistemas, visando apresentar aos usuários suas vantagens e limitações certamente contribuirá para melhorar o nível de utilização dos sistemas e/ou nortear sua evolução.

4.2 Algumas abordagens de Engenharia de Requisitos para Projetos de Telessaúde/ Telemedicina

Aqui nesta seção estaremos discutindo a abordagens de Engenharia de Requisitos em projetos de desenvolvimento de software para Telemedicina e Telessaúde, pesquisadas em trabalhos em andamento de grupos de pesquisa distintos, em comparação com experiências vivenciadas pelo autor recentemente neste contexto.

4.2.1 Método Volere para Estudo de Requisitos em Softwares Embarcados para Telemedicina

A aplicação do Método Volere [41], por Kondo [29], do Laboratório de Sistemas Integráveis do Núcleo de Telemedicina da Escola Politécnica da USP, em projetos de desenvolvimento de Software embarcado no segmento da Telemedicina, foi uma referência interessante encontrada em nossa pesquisa.

O problema a ser estudado é a aplicabilidade do software embarcado na Telemedicina, focando num ambiente humano-computador interoperável, de baixo custo e com mínimo de falhas. Outros importantes fatores a serem adicionados: alta performance e pervasivo. Buscou-se neste trabalho focalizar o problema através de duas lentes de aproximação: a lente da Telemedicina e a lente da Tecnologia da Informação.

A modelagem dos Requisitos dentro da Telemedicina poderia ser concebida sob qualquer um dos modelos citados na seção do Capítulo 3.2. Na seleção do método, no entanto, deverão ser considerados os seguintes pontos: um método que seja bem documentado, adaptável ao domínio da Telemedicina, uma técnica específica a este domínio, e que tenha uma seqüência bem definida e criteriosa.

Assim, segundo Kondo [29], o método **Volere** seria um dos métodos que obteria maior aderência aos pontos citados.

O Método Volere é um roteiro bem estruturado e completo para obtenção de requisitos. Baseia-se nos casos de uso, tendo como finalidade conduzir ao entendimento de forma mais abrangente, do domínio estudado e foi o que melhor correspondeu às necessidades. O esquema simplificado do método é mostrado na Figura 6. O caso de uso descreve a parte do serviço, do ponto de vista do usuário. Os requisitos funcionais descrevem as ações pertinentes ao sistema. Os requisitos não-funcionais descrevem a experiência obtida pelo usuário durante a utilização do sistema, ou seja, os requisitos não-funcionais são as propriedades ou qualidades que o sistema computacional deve conter que o tornam atrativo e próprio ao uso. Os requisitos não-funcionais não definem ou alteram a funcionalidade do produto, mas conferem a este a característica que corresponde à impressão causada ao usuário pelo seu uso.



Figura 6 – Esquema do Método Volere. Fonte: Adaptado de Robertson [41].

O Portal de Qualidade do Método Volere aplica-se aos requisitos obtidos realizando uma série de testes, que permitem testar conflitos e verificar se o requisito atende às especificações do sistema. Este permite que cada requisito tenha completude, rastreabilidade, consistência, relevância, corretude, ambigüidade e viabilidade. Os requisitos são testados contra as restrições globais do sistema, e é feito um cruzamento requisito a requisito verificando incompatibilidades.

Em geral, são verificados conflitos entre requisitos solicitados por diferentes stakeholders. O resultado final da aplicação da técnica é o Documento de Requisitos de Software.

Durante o processo de elicitação de requisitos é preciso representar o conhecimento de forma estruturada, por conta das indefinições do início de qualquer projeto, se torna muito mais nebuloso quando tratamos de um contexto como este. De um lado, a diversidade de artefatos envolvidos no universo de software embarcado e, por outro lado, pela riqueza de tipos de stakeholders envolvidos no processo de elicitação de requisitos. Podemos utilizar o modelo de processo de requisitos Volere para a elicitação de requisitos. Este é um modelo de processos genéricos, necessários para elicitar, especificar e rever os requisitos. É um guia de como descobrir e coletar os requisitos a serem validados. O modelo foca em conteúdo, ou seja, as dependências entre os processos que são definidos pelas interfaces. Este modelo não implica em nenhuma seqüência, e traz o detalhamento de todas as atividades e conexões entre eles (ver Figura 6). No que tange à fase de definição dos stakeholders, é importante uma avaliação criteriosa para identificar as pessoas que vão ajudar a especificar os requisitos. Adicionalmente, valemos dos dados estatísticos populacionais para conhecermos os grupos potenciais de stakeholders da base populacional envolvidos. Assim, agregaremos os elementos estatísticos para complementar o processo.

O protótipo, no entanto, não foi testado pelos stakeholders, o que traria sem dúvidas benefícios à análise. A partir de sugestões e críticas levantadas. No entanto, ocasionaria uma nova rodada em todas as etapas da Análise de Requisitos.

4.2.2 Cenários, Casos de Uso e UML para desenvolvimento de um Sistema Integrado de apoio ao Telediagnóstico e à Segunda Opinião Médica: HealthNet

A utilização da técnica de Cenários [28], Casos de Uso e UML [25] para elicitação, análise e modelagem dos requisitos de um Sistema Integrado de apoio ao telediagnóstico e à segunda opinião médico [34, 36, 37], desenvolvido pelo Grupo de Pesquisa de Tecnologia da Informação em Saúde – TIS da UFPE é uma das abordagens clássicas para Engenharia de Requisitos, na área de Telemedicina e Telessaúde.



The image shows a screenshot of a web browser displaying a form titled "FORMULÁRIO DE PARECER (LAUDO) DE NEFROLOGIA". The form is part of the HealthNet system, which is associated with the Universidade Federal de Pernambuco (UFPE) Hospital das Clínicas. The form is divided into two main sections: "01 DADOS DA SOLICITAÇÃO" and "02 DADOS DO PACIENTE".

01 DADOS DA SOLICITAÇÃO

Responsável:	Lais de Mendonça Neves	Cremepe:	441254542
Instituição/unidade:			
E-mail:	lais_mendonca@hotmail.com		

02 DADOS DO PACIENTE

Registro:			
Nome:	Jorge Danilo Emerenciano Coelho		

Figura 7 – Formulário de solicitação de biópsia e laudo no Sistema HealthNet. Fonte: Mello [37].

Alexandre José Henrique de Oliveira Luna

A versão 1.5 do HealthNet, em discussão, possibilita, aos profissionais de saúde da rede de atenção básica, solicitar, a profissionais especialistas de centros de referência, apoio no diagnóstico e conduta terapêutica de seus pacientes através da internet. Esta ferramenta vem sendo utilizada pelos parceiros da Rede de Núcleos de Telessaúde (NUTES) de Pernambuco, gerenciada pelo NUTES/HC-UFPE.

O desenvolvimento deste trabalho foi estruturado em três macro-atividades:

- **Análise e especificação de requisitos** – foram realizadas reuniões com a equipe da Unidade de Cardiologia e Medicina Fetal (UCMF) do Real Hospital Português. Os requisitos do sistema foram levantados e especificados de acordo com: (i) as necessidades levantadas pelos médicos, (ii) o material bibliográfico pesquisado que relata experiências e considerações no desenvolvimento de sistemas de Telemedicina e (iii) uma aplicação de telediagnóstico existente e atualmente em fase de testes na UCMF.
- **Projeto do sistema** – baseando-se nas experiências de sistemas de Telemedicina existentes, na avaliação das tecnologias da informação e no estudo de metodologias de projeto, o sistema foi projetado e documentado utilizando a notação UML (Unified Modeling Language), linguagem de modelagem padrão adotada pela OMG (The Object Management Group) [45]. Nesta etapa, as tecnologias para a implementação que se adequaram aos objetivos do sistema foram estabelecidas.
- **Implementação** – um protótipo do sistema foi construído a fim de validar a proposta do modelo do HealthNet. Este protótipo foi adaptado para fornecer serviços de Telemedicina na área de cardiologia fetal e pediátrica. A escolha desta área ocorreu devido à sua importância na saúde pública e ao perfil dos médicos colaboradores deste projeto.

Alexandre José Henrique de Oliveira Luna

Este trabalho é considerado multidisciplinar, visto que além do conhecimento das necessidades da área de saúde, um estudo em diversas áreas da Ciência da Computação foi efetuado para dar suporte à análise, projeto e implementação do HealthNet.

4.2.3 Prototipação Evolutiva e SCRUM para desenvolvimento do Módulo de Biblioteca Multimídia do dataNUTES: uma experiência de desenvolvimento distribuído de software

Como atividade parcial para conclusão da disciplina de Engenharia de Software, da grade curricular Mestrado em Ciência da Computação do Centro de Informática (CIn), da Universidade Federal de Pernambuco (UFPE), foi realizada a proposta para criação de uma fábrica de software, visando o desenvolvimento distribuído de um produto de software [5].

Os professores da disciplina Engenharia de Software disponibilizaram quatro fábricas entre os pós-graduandos do Centro de Informática da UFPE, e informaram os papéis básicos, necessários ao funcionamento e operação de cada empresa de software, com finalidade de criar equipes multidisciplinares, estabelecendo critérios de habilidades e responsabilidades.

Realizada a escolha do “cliente” a ser atendido pela fábrica de software, através de uma “audiência pública”, onde vários projetos de várias instituições, incluindo a Universidade Federal Rural de Pernambuco (UFRPE), o Centro de Estudos e Sistemas Avançados do Recife (C.E.S.A.R.) e o próprio CIn-UFPE, foram apresentados como opção para aplicação do experimento da disciplina; o Núcleo de Telessaúde (NUTES) do Hospital das Clínicas (HC) da Universidade Federal de Pernambuco (UFPE), sagrou-se vencedor através da proposta de construção uma

Alexandre José Henrique de Oliveira Luna

solução de software para armazenamento, conversão e recuperação inteligente de conteúdos multimídia, instrumentos de telessaúde/telemedicina como uma alternativa para melhorar o acesso à saúde para a população carente, em zonas rurais ou desprovidas de serviços médicos especializados, promovendo o cuidado à saúde através de programas de educação e prevenção ao nível da atenção primária, denominada Biblioteca Multimídia dataNUTES (BMD) [5, 33].

Para esta aplicação, o cliente solicitou recursos de servidor de Streaming, com Player Web, sistema de busca, além da compatibilidade com os sistemas nativos (ambiente de ensino e aprendizagem) disponível nos servidores do cliente, de forma a não depender de Sistema Operacional, ou necessidades de instalações de outros softwares proprietários.

A alocação dos integrantes a cada fábrica de software se deu pela livre escolha dos alunos, preenchendo o quadro de vagas disponíveis, sem qualquer questionamento ou mesmo interferência na re-alocação de pessoal, deixando tal responsabilidade a cargo de cada fábrica. A partir de estruturação da equipe, os seus componentes foram re-allocados, após identificação das habilidades, de acordo com o perfil que melhor se adequava às necessidades da fábrica de software. A esta equipe foi atribuída, após reunião inicial de concepção do empreendimento, o nome ***factory2u.net***.

A ***factory2u.net***, alusão ao posicionamento na matriz de alocação dos membros da fábrica de software número dois, cujo endereço na Internet, faz referência ao próprio nome da fábrica, conforme citado e em negrito. Foi utilizado o SCRUM de forma customizada e otimizada para particularidades existentes em seu dia-a-dia, buscando sempre preservar a essência desta abordagem.

Alexandre José Henrique de Oliveira Luna

Para o desenvolvimento e acompanhamento dos processos, bem como para a evolução da codificação dos requisitos da Biblioteca Multimídia, apresentadas pelo cliente NUTES, foi escolhida dentre alguns métodos ágeis, o SCRUM [31], por apresentar algumas vantagens, como possibilitar um rápido entendimento das necessidades do cliente, a disponibilidade do cliente, o qual é integrante da equipe e analista de sistemas do cliente, representando-o nas decisões e encaminhamentos.

Para viabilidade da construção da Biblioteca Multimídia foi realizado um amplo levantamento bibliográfico, além de pesquisa qualitativa que elucidaram as principais características e ferramentas de manipulação de vídeo, necessárias ao desenvolvimento, a fim de proporcionar entendimento técnico e agregar habilidades à equipe, para o trabalho com esses recursos.

Para o processo de Engenharia de Requisitos do Projeto, com base no SCRUM [31], foi adotado o processo de Prototipagem Evolutiva, mas que em muitos dos casos terminou sendo descartável. Com base no esboço das telas, realizados juntos aos stakeholders: profissionais de saúde, Gerente de Videoconferência, Gerente de Sistemas, Coordenação do NUTES/UC-UFPE, etc. Posteriormente estas interfaces eram aperfeiçoadas, pela Equipe de Business Design da Fábrica, liderada pelo autor, que também exerceu o papel de Product Owner no ciclo de vida do SCRUM.

Alexandre José Henrique de Oliveira Luna

Após o processo de refinamento das telas, estas eram, enfim, novamente submetidas à validação pelos clientes. Quando, o conjunto de telas de uma Sprint havia sido homologada plenamente pelos stakeholders envolvidos, a Equipe de Business Design elaborava as “estórias” que serviriam de base para popular o Product Backlog, a partir daí o processo de Planning Poker para definição da complexidade de implementação de cada estória era jogado. O processo, então, se repetia para cada Sprint do Projeto.

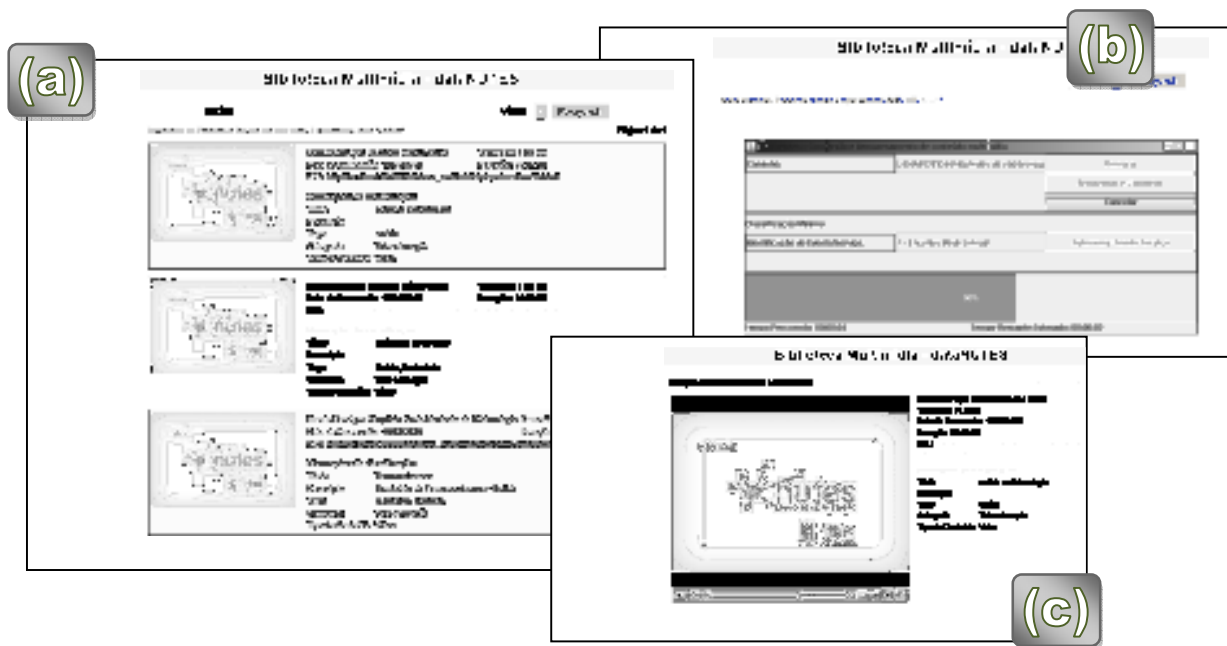


Figura 8 – Telas do Módulo de Biblioteca Multimídia do dataNUTES - (a) Resultados do sistema de busca por palavras-chave; (b) O módulo Desktop realize o upload da mídia, a ser classificada e publicada na Web, para o servidor de Streaming do dataNutes; (c) Exibição do vídeo em modo Streaming, através do Player Web. Fonte: LUNA [5].

A abordagem de prototipação no processo de Engenharia de Requisitos foi essencial para o sucesso da plena aceitação dos produtos de softwares gerados por este projeto, por parte dos stakeholders envolvidos.

O processo baseado em SCRUM, numa primeira análise extremamente simples (sem muita burocracia e com pouca documentação), se demonstrou bastante eficaz, gerando os resultados desejados. O site da fábrica

Alexandre José Henrique de Oliveira Luna

(www.factory2u.net) se apresentou com o uma eficaz ferramenta de comunicação, embora tenha havido, em algumas circunstâncias, certo delay de sincronização das informações em função do trabalho distribuído da Equipe.

Foi necessário renegociar escopo em função da produtividade variável da Fábrica, constatadamente em função da dedicação, também variável de seus membros, em função dos demais compromissos pessoais e profissionais de cada um. Em função disso ocorreu uma redução de um terço do escopo original, bem como uma proposição de reavaliação de tecnologia: desenvolvimento da camada Web em Python, em detrimento do aumento de produtividade de implementação. O cliente participou ativamente destas negociações e assumiu o risco de legado deste projeto, isso apesar de não possuir, atualmente, recurso interno com expertise para manter o código posteriormente.

Apesar de ter ocorrido um número razoável de "não conformidades" nos testes da Sprint 2, o que levou a Fábrica a ter um overhead de retrabalho, durante a realização das Sprints posteriores, os resultados do projeto foram extremamente satisfatórios tando do ponto de vista da oportunidade da aplicação do conhecimento de Engenharia de Software recebido e aperfeiçoado como proposta da disciplina IN953 – Engenharia de Software Distribuída, da grade Curricular do Mestrado em Ciências da Computação do Centro de Informática (CIn) da Universidade Federal de Pernambuco (UFPE); quanto do ponto de vista da geração de uma contribuição concreta ao Cliente do projeto (NUTES), na elaboração e construção de uma ferramenta de software para compartilhamento de instrumentos de Telemedicina/Telessaúde como alternativa de melhoria do acesso à saúde, e para a população carente, em zonas rurais ou desprovidas de serviços médicos especializados, promovendo o cuidado à saúde através de programas de educação e prevenção ao nível da atenção primária, que são metas do Projeto REDE NUTES.

Embora o resultado global tenha sido positivo, este tipo de experiência com um modelo de gerenciamento com base no voluntariado, não seria aplicável em projetos comerciais sem algum tipo de regulamentação sobre remuneração/premiação por produtividade, uma vez que em algumas situações o bem comum da Equipe gerou a necessidade de atos heróicos de integrantes mais comprometidos, em função da temporária indisponibilidade ou capacidade de dedicação de outros membros, numa fase crítica do projeto. Mesmo reconhecendo, para o caso em estudo, que o cliente não tenha sido prejudicado no resultado final do projeto, os indivíduos que mais se dedicaram mereceriam uma distinção em função da superação a que se submeteram.

Capítulo 5

5 Conclusão e considerações finais

Como aprendizado proporcionado pela oportunidade de realizar esta ampla revisão na literatura científica a respeito de abordagens de Engenharia de Requisitos para Projetos de Desenvolvimento de Software no domínio de Telemedicina e Telessaúde, podemos destacar, a oportunidade de avaliação das técnicas, métodos e experiências aplicadas no contexto de projetos desta natureza, bem como a exploração dos aspectos que tornam particulares a abordagem de Engenharia de Requisitos para desenvolvimento de softwares nesta área do conhecimento.

A abordagem de métodos e técnicas de especificação formal para Engenharia de Requisitos ainda é classicamente empregado para abordagem de sistemas de telemedicina e telessaúde [29, 36 e 37], no entanto o advento de métodos ágeis, como o SCRUM [25], vem propondo uma alternativa menos formal para aplicação dos processos de Engenharia de Requisitos em projetos desta natureza, com sucesso. Vimos que no caso do projeto da Biblioteca Multimídia do dataNUTES [5] a aplicação de prototipação no processo de Engenharia de Requisitos foi essencial para o sucesso dos produtos de software gerados.

Apesar da abordagem simples sob o ponto de vista do SCRUM [31], com uma documentação de requisitos minimalista, baseada em: protótipos das interfaces a serem construídas e descrição simples, mas objetiva das histórias a serem implementadas, como *instrumentos centrais* dos processos de Elicitação, Análise, Negociação, Documentação e Validação dos Requisitos do Projeto, a riqueza dos detalhes do processo de desenho das interfaces fez a diferença traduzindo bem o contexto de que “uma imagem vale mais que mil palavras”. Sendo assim podemos dizer que, seguramente, os resultados deste trabalho foram bastante satisfatórios nos contextos: do produto de software gerado, da primordial abordagem do processo de Elicitação de Requisitos, baseada na técnica de Prototipação e

na aplicação de uma abordagem de fábrica de software num processo de desenvolvimento distribuído de software, com foco no aprendizado dos conhecimentos abordados pela disciplina.

É de notório conhecimento que mais da metade dos defeitos de um software possuem suas raízes nos requisitos [2, 28], seguidos pelos erros de design, erros de código aparecem como a terceira maior causa de problemas e erros diversos correspondem ao restante dos problemas. Também deve ser considerado o alto custo para se corrigir os erros do software que possuem origem nos requisitos, o custo para se corrigir um erro dessa natureza é muito maior que o custo para se corrigir um erro de design ou código.

Softwares relacionados à área de saúde, telessaúde e telemedicina precisam ter seus requisitos melhor definidos e implementados que softwares de outra natureza, em vista de uma série de fatores críticos, como: sua disponibilidade para sistemas de missão crítica no monitoramento de sistemas de suporte à vida; nos riscos envolvidos na delicada área de sua aplicação, como é o caso do tratamento de sistemas onde o tempo de resposta é um fator crítico; nos aspectos de confidencialidade envolvidos em relação aos dados clínicos de paciente; ou mesmo na confiabilidade das informações mantidas pelo software, evitando inconsistência nos dados dos pacientes, sendo imprescindível no apoio ao tele-diagnóstico; ou ainda permitindo a usabilidade ideal para o uso a que se destina, através de uma interface de extrema facilidade de uso e ergonomia adequados para o uso por diversas horas a fio por profissionais de saúde no desafio de salvar vidas.

Enfim, podemos concluir que o processo de Engenharia de Requisitos bem estruturado, planejado e executado pode não garantir a construção de softwares de saúde, telessaúde e telemedicina de qualidade, mas a ausência de um processo de Engenharia de Requisitos bem definido e executado, garante o insucesso de qualquer projeto de software neste contexto, principalmente nestas áreas em função de suas características diferenciadas e do acesso a profissionais da área.

Referências

6 Referências

- [1] World Health Organisation – WHO/OMS, <http://www.who.int>, acessado em 03/07/2008;
- [2] MIYAZAKI, M., Future of Telemedicine/ Telehealth/e-Health, Business Briefing: Hospital Engineering & Facilities Management, <http://devserver3.devstudios.com/ettri/admin/PDFs/12.pdf>, acessado em 03/07/2008.
- [3] Jim Johnson. 1994. *Chaos: Charting the Seas of Information Technology*. Published Report. The Standish Group.
- [4] DIAS, F. da C ; CASTRO, J. B de; *O Processo de Engenharia de Requisitos para Softwares aplicados a Saúde*; Monografia apresentada como requisito parcial para conclusão da Disciplina IN1020 – Engenharia de Requisitos do Curso de Mestrado em Ciência da Computação do Centro de Informática (CIn) da UFPE; Recife, 07/04/2008;
- [5] LUNA, Alexandre J. H. de O; ARAÚJO, Ana C. M. L.; SILVA, Haglay A. N.; PESSOA, João T. F. S.; ALBUQUERQUE, Jones O.; ARAÚJO, Raony M.; PATRÍCIO, Raphael F. A.; MORAES, Marcela B. S.; ANDRADE, Maria S. B.; MENDONÇA, Sérgio F. T. O.; MEIRA, Sílvio R. L.; *Desenvolvimento distribuído do Módulo Biblioteca Multimídia do dataNUTES com a Metodologia Ágil SCRUM*. Relatório de Projeto apresentado como requisito parcial para conclusão da Disciplina IN953 – Engenharia de Software do Curso de Mestrado em Ciência da Computação do Centro de Informática (CIn) da UFPE; Recife, 04/07/2008;
- [6] Crowston, K.; Annabi, H.; Howison, J.: Defining Open Source Software Project Success, In Proceedings of the 24th International Conference on Information Systems (ICIS), 2003.
- [7] Leão, B. de F., Padrões HL7 Introdução a Padrões e Interoperabilidade, 2007.
- [8] Houaiss, Antônio; Villar, Mauro de Salles; Franco, Francisco Manoel de Mello; *Dicionário Houaiss da Língua Portuguesa*; Objetiva, Rio de Janeiro, 2001;
- [9] Ferreira, Aurélio Buarque de Holanda; *Dicionário da Língua Portuguesa*; 5ª Edição revisada e ampliada; Nova Fronteira; Rio de Janeiro, 2001;
- [10] Shortliffe, E.H., Perreault, L.E., Wiederhold, G. and Fagan, L. (eds.) 1990, *Medical Informatics: Computer Applications in Health Care*, Menlo Park, Addison Wesley.
- [11] Sociedade Brasileira de Informática em Saúde – SBIS; <http://www.sbis.org.br>; acessado em 03/07/2008;
- [12] University Health Network – UHN (CAN), http://www.uhn.ca/Clinics & Services/telehealth/what_is_telehealth.asp, acessado em 04/07/2008;

- [13] Health Telematics Unit from University of Calgary (CAN);
http://www.fp.ucalgary.ca/telehealth/What_Is_Telehealth.htm, acessado em 04/07/2008;
- [14] Conselho Brasileiro de Telemedicina e Telessaúde; <http://www.cbtms.com.br/>; acessado em 28/06/2008;
- [15] Site <http://www.urcamp.tche.br/ccei/revista15.pdf>, acessado em 04/02/2008;
- [16] Site <http://www.estacaodigitalmedica.com.br/telemedicina>, acessado em 03/09/2007;
- [17] Site do British Medical Journals; <http://bmj.bmjournals.com/cgi/content/full/323/7312/557>, acessado em 04/07/2008;
- [18] Emergency Medicine and Primary Care Home Page; <http://www.embbs.com>; acessado em 14/03/2008;
- [19] Site ISO (www.iso.org), acessado em 15/01/2008.
- [20] Site ANSI (www.ansi.org), acessado em 15/01/2008.
- [21] Jr., F. P. B., No Silver Bullet.: Essence and Accidents of Software Engineering, 2003.
- [22] Sommerville, Ian; SAWYER, Pete. Requeriments Engineering. A Good Praticce Guide. England: John Wiley e Sons Lid, 1997.
- [23] S. M. Easterbrook, "What is Requirements Engineering?" (Draft book chapter), acessado em 25/02/2008. Disponível em: <http://www.cs.toronto.edu/~sme/papers/index.html>.
- [24] Site do IEEE, disponível em <http://www.ieee.org/portal/site>, acessado em 25/02/2008.
- [25] Larman, Craig. Applying UML and patterns: an introduction to object-oriented analysis and design and the Unified Process. – 2nd ed. – New Jersey: Preintice Hall PTR, pp. 41-44, 2001.
- [26] Takeuchi, H. and I. Nonaka, The New New Product Development Game. Harvard Business Review, 1986 (January-February).
- [27] Schwaber, K., Agile Project Management With SCRUM, Microsoft, 2004.
- [28] PRESSMAN, Roger. Software Engineering: A Practitioner's Approach, 6ª edição, Mc Graw Hill, 2005.
- [29] Kondo, Márcia N.S.; Silva, José R. da; Hira, Adilson Y.; Zuffo, Marcelo K.; *Estudo de Requisitos do Software Embarcado no Segmento da Telemedicina*; Núcleo de Telemedicina, Laboratório de Sistemas Integráveis, Escola Politécnica a Universidade de São Paulo - USP, 2006;
- [30] Rocha, H.V., Baranaukas, M.C.C. (2000) Design e avaliação de interfaces humanocomputador. Escola de Computação 2000. Departamento de Ciência da Computação IME- USP. São Paulo.
- [31] MOUNTAIN Goat Software, The Scrum Development Process, Disponível em <http://www.mountaingoatsoftware.com/Scrum> [Junho, 2006].
- [32] Novaes M A; Araújo K S; Couto J M L A. A Experiência de Pernambuco em telessaúde. In: Alaneir de Fátima dos Santos; Cláudio de Souza; Humberto José Alves; Simone Ferreira dos Santos. (Org.). Telessaúde Um Instrumento de Suporte Assistencial e Educação Permanente. 1 ed. Belo Horizonte: Editora UFMG, 2006, p. 198-233.

- [33] J. O. A. Segundo, C.H.L.Souza, e M.A.Novaes, "Biblioteca Digital de Videoconferências: Uma Estratégia para Ampliar o Acesso a Programas de Capacitação para o PSF", CBIS 2006, Florianópolis, 14-18 de outubro de 2006, p.1255-1260.
- [34] M.R.Mello e M.A.Novaes, "Evolving a Second Opinion Software to Web 2.0", Medinfo 2007, IOS Press, Brisbane – Australia, 20-24 de agosto de 2007, p. 135.
- [35] Portal do Projeto RedeNUTES, disponível em <<http://www.redenutes-pe.ufpe.br/>>. Acesso em 26 de março de 2008.
- [36] M.R. Mello; M.E.M. Filho; L.M. Neves; D. V. da Silva; M. A. Novaes; *Estratégia para Avaliação de um Sistema de Cooperação em Saúde na Web*; NUTES/HC-UFPE, Recife, 2006;
- [37] M.R. Mello; M.E.M. Filho; L.M. Neves; D. V. da Silva; M. A. Novaes; *Utilização de um Sistema de Cooperação na Web para Solicitação e Emissão de Laudos de Nefropatologia*; NUTES/HC-UFPE, Recife, 2007;
- [38] H. Ledang. Formal Techniques in the Object-Oriented Development: an Approach based on the B method. In the 11th PhDOOS Workshop: PhD Students in Object-Oriented Systems, Budapest (H), <http://www.st.informatik.tudarmstadt.de/phdws>
- [39] Mizuno S., Akao Y.. QFD: The Customer-Driven Approach to Quality Planning and Deployment. 1994. Translated by Glenn Mazur.
- [40] MURATA, T. Petri net: properties, analysis and applications. Proceedings of the IEEE, v. 77, n. 4, p. 541-579, 1989.
- [41] Robertson, Suzanne and James. Mastering the requirements process. London: ACM Press, 1999.
- [42] M. L. Moreira, L. S. Vidal, J. A. Placca, C. M. de O. Perez, S. H. de Souza; GERENCIAMENTO DE DOCUMENTAÇÃO E REQUISITOS A PARTIR DE NORMATIVOS: UMA EXPERIÊNCIA NA ANS; Coordenação de Informações, Diretoria de Normas e Habilitações de Produtos, Agência Nacional de Saúde Suplementar - ANS, Brasil, 2006;
- [43] Littlejohns, P.; Wyatt, J. C., Garvican, L.; "Evaluation computerised health information systems: hard lessons still to be learnt", British Medical Journal, ISSN 0959-8146, vol 326, n. 7394, p. 860-863, April, 2003;
- [44] NORMAN, Donald A.; The Design of Everyday Things. New York (New York), Currency Doubleday. 1990. 257p.
- [45] Site da Object Management Group - OMG; <http://www.omg.org/>; acessado em 06/07/2008;