

DANIEL BONFIM DA SILVA
RUBENS JUN HIRAKAWA
SÉRGIO ENRIQUE GARBATI KREINER

EXTREME PROGRAMMING (XP)
APLICAÇÃO E ANÁLISE

Monografia apresentada à Escola
Politécnica da Universidade de São Paulo
para conclusão do Curso de Graduação
em Engenharia da Computação

São Paulo
2005

DANIEL BONFIM DA SILVA
RUBENS JUN HIRAKAWA
SÉRGIO ENRIQUE GARBATI KREINER

EXTREME PROGRAMMING (XP)
APLICAÇÃO E ANÁLISE

Monografia apresentada à Escola
Politécnica da Universidade de São Paulo
para conclusão do Curso de Graduação
em Engenharia da Computação

Áreas de Concentração:
Processos de Software
Engenharia de Software

Orientador:
Prof^ª. Dr^ª. Selma Shin Shimizu Melnikoff

São Paulo
2005

AGRADECIMENTOS

À Prof^a. Dr^a. Selma Shin Shimizu Melnikoff, pelo apoio, serenidade e disposição para guiar este trabalho ao rumo certo e por compartilhar com seus alunos orientados seu conhecimento e experiência.

E a todos os que participaram ou contribuíram com o planejamento e o desenvolvimento deste trabalho.

RESUMO

Este trabalho tem, como objetivo, apresentar a aplicação de *Extreme Programming* (XP) a um projeto de software real e relatar os pontos relevantes desta experiência; Inicialmente descreve a metodologia XP e as características do sistema de apoio médico, cujo desenvolvimento foi realizado através do XP, utilizando as suas práticas e os seus valores. Em seguida são relatados os resultados da aplicação do XP e as alternativas utilizadas para adequação das suas práticas às necessidades reais do projeto desenvolvido.

ABSTRACT

This research considers a study of the suitability of Extreme Programming (XP) to a real software project; it is composed of a complete description of XP methodology, an experiment in using XP practices and values for the planning and development of a medical support system and, after all, the results of using XP and some ways to let it fit to the real project needs.

SUMÁRIO

1	Introdução	4
1.1	Considerações iniciais	4
1.2	Objetivo	5
1.3	Motivação	6
1.4	Estrutura da monografia	7
2	<i>Extreme Programming</i>	9
2.1	Os valores do XP	11
2.2	Princípios básicos do XP	12
2.3	Divisão do trabalho no XP	14
2.4	Práticas do XP	15
2.5	Ciclo de vida de um projeto XP ideal	16
3	Descrição do sistema	18
4	Aplicação do XP	21
4.1	Processo Aplicado	21
4.2	Divisão de trabalho na equipe	24
4.3	Participação do cliente	25
4.4	Story Cards e Task Cards	26
4.5	Planning Game (iterações e prioridades)	27
4.6	Sessões de programação	28
4.7	Testes	28
4.8	Comunicação	28
4.9	Padrões de programação	29
4.10	Sistema de apoio ao projeto	29
5	Avaliação da aplicação	32
5.1	Equipe	32
5.2	Escolha de prioridades	33
5.3	Iterações	34
5.4	Programação em pares	34
5.5	Testes funcionais	36
5.6	Reuniões	36
5.7	Padrões de programação	37
6	Considerações finais	38
6.1	Conclusão	38
6.2	Contribuições	40
6.3	Trabalhos futuros	40
7	Bibliografia	41
8	Anexos	42
8.1	Especificação de Requisitos de Software – SPM	42
8.2	Especificação de Requisitos de Software – SPW	42
8.3	Relatório de Produtividade – Projeto XP	42

ÍNDICE DE FIGURAS

Figura 1: Arquitetura do sistema	18
Figura 2: Processo aplicado	21
Figura 3: Detalhamento da atividade iterativa	22
Figura 4: Produtos resultantes do planejamento	23
Figura 5: Produtos resultantes das sessões de programação	23
Figura 6: Produtos resultantes do <i>productionizing</i>	24
Figura 7: Sistema de apoio ao projeto – tela principal	30
Figura 8: Sistema de apoio ao projeto – tela de rendimento	31

NOTA

Os termos em *itálico* deste texto foram mantidos no idioma inglês, como na bibliografia consultada, por apresentarem tradução conflitante, ou por serem conhecidos através de siglas compostas por suas iniciais em inglês, ou pelo uso corrente da nomenclatura no idioma original.

1 Introdução

Neste capítulo serão abordadas a origem do *Extreme Programming*, junto com as demais metodologias ágeis, a motivação deste trabalho, embasada no atual contexto do desenvolvimento de software e, por fim, uma descrição estrutural desta monografia.

1.1 Considerações iniciais

De 11 a 13 de fevereiro de 2001, encontraram-se em uma estação de esqui em Utah, nos Estados Unidos, 17 pessoas que tinham como objetivo formalizar as idéias que compartilhavam sobre um modo novo de se desenvolver software. Representantes do *Extreme Programming* (XP), Scrum, *Dynamic Systems Development Method* (DSDM), *Adaptative Software Development* (ASD), *Crystal Methods*, *Feature-Driven Development* (FDD), *Pragmatic Programming* e outros simpatizantes de alternativas ao desenvolvimento de software rigoroso e orientado à documentação compuseram o grupo (HIGHSMITH, 2003).

Participaram deste encontro Kent Beck, criador do XP, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Kent Schwaber, Jeff Sutherland e Dave Thomas (AGILE MANIFESTO, 2001).

O encontro produziu o Manifesto para Desenvolvimento Ágil de Software, em que foram especificadas as prioridades das metodologias ágeis (HIGHSMITH, 2003):

- Indivíduos e interação, frente aos processos e ferramentas: pois os projetos são movidos por indivíduos e a conversação entre os membros da equipe é a maneira mais eficiente de se trocarem informações;
- Software em funcionamento, frente à documentação: pois o sistema de software em funcionamento é a principal evidência de progresso no projeto;

- Colaboração com o cliente, frente à negociação de contratos: pois a prioridade do Desenvolvimento Ágil de Software é satisfazer o cliente através de entregas pontuais e contínuas de software com alto valor agregado;
- Resposta a mudanças, frente ao seguimento de um plano: pois a aceitar mudanças em um projeto de software resulta no ganho de vantagem competitiva.

A Aliança Ágil, grupo de idealizadores e praticantes de metodologias ágeis, foi firmada em 2001, mas a ideologia por ela adotada havia começado há 15 anos; neste período os autores do Manifesto já haviam começado a praticar suas idéias em projetos reais (HIGHSMITH, 2003).

Atualmente, o *Extreme Programming*, ou *XP*, é a metodologia ágil mais conhecida; uma possível razão para isto é o embasamento prático deste modo de desenvolvimento de software: o XP surgiu da experiência de Kent Beck e Ward Cunningham que trabalharam juntos em diversos projetos e avaliaram as práticas de programação e gestão que davam melhores resultados. Estas serviram como base para o XP (SHARMA, 2004).

A escolha do XP como metodologia de um projeto de software, embora tenha como garantia a experiência de seus criadores, depende de um grande entrosamento entre os membros da equipe, que muitas vezes não se enquadra ao comportamento e cultura destes (STEPHENS, ROSENBERG, 2003), e de práticas que, à primeira vista, podem não agradar os programadores, como refazer o mesmo código inúmeras vezes até que se atinja o resultado mais simples e eficaz. Portanto, alcançar a eficiência num projeto XP exige bom senso e a realização de mudanças culturais significativas na equipe.

1.2 Objetivo

Este trabalho tem como objetivo avaliar cada um dos valores e práticas do XP, quanto à viabilidade da sua aplicação e à eficiência que agrega ao desenvolvimento, em um caso real, do sistema de apoio médico, além de propor alternativas às práticas inadequadas ao

projeto em questão. Tal estudo visa a comprovar o valor do XP como metodologia de desenvolvimento e propor alternativas para seus pontos críticos.

1.3 Motivação

Atualmente, sistemas de software são produtos consolidados no mercado. A NASDAQ (*National Association of Securities Dealers Automated Quotation System*), bolsa de valores eletrônica dos Estados Unidos, tem mais de 4.800 empresas listadas, sendo que destas 297, ou seja, 6,2% produzem software (NASDAQ, 2005).

Ainda que no Brasil não se negociem tantas ações de empresas de software como nos Estados Unidos – só a Ideiasnet S.A. está registrada na Bovespa (Bolsa de Valores de São Paulo) como empresa de software (BOVESPA, 2005) – há no país importantes empresas neste setor. A ABES (Associação Brasileira das Empresas de Software) tem mais de 700 empresas associadas; elas, juntas, faturam 3,2 bilhões de dólares anualmente (ABES, 2005).

Nos últimos anos, os sistemas de software deixaram de ocupar apenas a função de suporte aos negócios e passaram a ser um ponto crítico para eles. São estes sistemas que levam ao desenvolvimento de novos produtos, de automóveis integrados por centenas de chips com software embarcado a celulares (HIGHSMITH, 2003).

"O futuro da nossa Era da Informação pertence ao Ágil, às companhias capazes de criar mudanças e até um pouco de caos para seus competidores" (HIGHSMITH, 2003, p.3). E no mercado de software, a agilidade é importante não só devido ao ganho de competitividade, mas também porque as necessidades do cliente mudam com o mesmo dinamismo que os negócios deste; da maneira como os sistemas de software são projetados atualmente, adaptá-los a mudanças é muito custoso (LAUDON, LAUDON, BRABSTON, 2002)

Dada a importância corrente do mercado de *software* e a necessidade de desenvolver este tipo de produto de modo ágil e adaptado a mudanças constantes de requisitos, optou-se neste trabalho pela aplicação do XP a um projeto de *software* e pela avaliação desta metodologia. O XP foi escolhido pois, embora se destaque frente às demais metodologias ágeis, carece de informações formais sobre seu desempenho.

1.4 Estrutura da monografia

Este trabalho tem a seguinte estrutura de capítulos:

- O capítulo 1, da Introdução, apresenta um breve histórico dos métodos ágeis e como, entre eles, apresenta-se o XP. Há ainda neste capítulo os objetivos deste trabalho e a motivação para seu desenvolvimento;
- O capítulo 2, intitulado *Extreme Programming*, traz uma descrição desta metodologia, desde seus valores e práticas principais, até a distribuição de tarefas entre os membros de uma equipe e o ciclo de vida ideal de um projeto XP;
- O capítulo 3, intitulado Descrição do Sistema, apresenta uma descrição resumida do sistema de apoio médico desenvolvido com uso do XP;
- O capítulo 4, intitulado Aplicação do XP, descreve como o Extreme Programming foi utilizado na gerência e no desenvolvimento do sistema descrito no capítulo 3, explicitando a adaptação dos valores e das práticas, onde foi necessário; são apresentadas a atribuição de papéis aos membros da equipe, a organização dos pares de programação, a aplicação e a organização das reuniões rápidas, e as maneiras, triviais ou adaptadas, de aplicar o XP no projeto;
- O capítulo 5, intitulado Avaliação da Aplicação, aborda o impacto que a aplicação do Extreme Programming causou no desenvolvimento do sistema. Neste capítulo são apresentadas, por exemplo, as mudanças de cronograma ocasionadas pelo refactoring e as características do XP que podem ser adaptadas ou aprimoradas para melhor atender o projeto;

- O capítulo 6, das Considerações Finais, é dedicado às conclusões obtidas com o trabalho desenvolvido, às contribuições que este traz em âmbito científico e prático, e aos trabalhos futuros que podem ser desenvolvidos nesta mesma linha de pesquisa;
- O capítulo 7 contém as referências bibliográficas em que se baseia este trabalho;
- Nos apêndices há um glossário e um documento de especificação de requisitos do sistema de software desenvolvido.

2 Extreme Programming

Extreme Programming (XP) é uma metodologia ágil para equipes pequenas ou médias desenvolverem software face a mudanças de requisitos rápidas ou indefinidas (BECK, 2000).

A metodologia tem em seu nome o termo *Extreme* pois se baseia em princípios e práticas de programação considerados produtivos por seus idealizadores (Kent Beck e Ward Cunningham) a níveis extremos. Estas práticas são (BECK, 2000):

- a revisão de código deve ser feita constantemente, durante a programação em pares (*pair programming*);
- a execução de testes deve ser feita continuamente pelos programadores (*unit testing*) e pelo cliente (*functional testing*);
- o design deve fazer parte do trabalho de todos (*refactoring*);
- o sistema deve ser o mais simples possível e suficiente para seu correto funcionamento;
- todos devem trabalhar definindo e redefinindo a arquitetura constantemente (metáfora);
- testes de integração devem ser feitos diversas vezes por dia (integração contínua);
- cada iteração deve durar apenas segundos, minutos e horas (*Planning Game*).

O XP se diferencia de outras metodologias devido a (BECK, 2000):

- sua realimentação constante e concreta: a cada reunião são apresentados o rendimento das sessões de programação, os problemas ou soluções importantes encontrados pelos programadores e as impressões do cliente em relação aos produtos parciais;
- seu planejamento incremental: o desenvolvimento é dividido em iterações cujo resultado são sistemas de software funcionais que, combinados, formam o produto final solicitado pelo cliente;

- possibilidade de agendar a implementação de novas funcionalidades de modo flexível: pois a resolução de problemas de integração e a otimização dos sistemas produzidos ocorrem durante o desenvolvimento e, portanto, são tratadas igualmente para os módulos previstos no planejamento inicial ou para os novos;
- dependência de testes escritos por programadores e pelo cliente para monitorar o progresso no desenvolvimento: os testes são consolidados como a forma oficial do XP para avaliar o progresso do desenvolvimento e para validar os sistemas produzidos;
- dependência da comunicação oral: que ocorre em reuniões constantes e nas sessões de programação e promovem a difusão de informações técnicas e gerenciais por toda a equipe;
- testes e código fonte para disseminação da estrutura do sistema e seu propósito: os testes escritos e o código fonte são as únicas documentações necessárias em um projeto XP; eles indicam todas as funcionalidades que devem ser validadas (testes) ou que foram implementadas (código fonte);
- dependência do projeto evolutivo do sistema: um projeto que use a metodologia XP deve ser evolutivo, ou seja, deve ter como resultado um sistema de software que ganha funcionalidades a cada término de iteração;
- colaboração entre programadores: os programadores trocam conhecimento técnico e atuam como revisores imparciais do código produzido durante as sessões de programação;
- combinação de práticas que se adequam tanto com os interesses de longo prazo do cliente quanto com os de curto prazo dos programadores: pois o a satisfação do cliente depende da entrega do produto final planejado e a dos programadores da execução com sucesso de programas simples. No XP, o produto final é uma combinação de vários programas simples, portanto a metodologia atende os interesses de toda a equipe.

2.1 Os valores do XP

Valores, no contexto do XP, são as premissas que motivam todos os princípios e práticas da metodologia.

Os quatro valores do XP são (BECK, 2000):

- Comunicação;
- Simplicidade;
- Realimentação;
- Coragem.

Estes valores se adequam tanto às necessidades humanas quanto às comerciais.

Quanto à comunicação, Beck (2000) menciona que problemas com projetos podem ser invariavelmente explicados porque os assuntos importantes não são comentados pelas pessoas. Algumas vezes, o programador não fala sobre uma mudança crítica no projeto, ou não faz a pergunta certa ao cliente e uma decisão importante sobre o domínio do sistema deixa de ser tomada. Outras vezes, o gerente não faz a pergunta certa ao programador e a evolução do sistema não lhe é totalmente revelada. A liberdade de comunicação com que um projeto feito usando-se XP deve ter reduz a chance de que informações importantes não cheguem ao conhecimento de toda a equipe; de qualquer forma, é atribuição do coach treinar as pessoas para que estas identifiquem e comuniquem assuntos e temas importantes para o projeto.

"XP aposta que é melhor fazer algo simples hoje e pagar no futuro para fazer as mudanças necessárias que fazer algo complexo, mas nunca utilizado" (BECK, 2000, p.30). A avaliação da relevância das funcionalidades previstas para um sistema de software é importante para garantir que, com o menor esforço possível, o sistema produzido tenha o maior valor agregado; um sistema composto apenas por funções essenciais tem valor muito maior, para o negócio a que se destina, que outro composto

por funções de pouca utilidade, embora ambos possam ter exigido o mesmo esforço para seu desenvolvimento.

A realimentação ocorre em diferentes escalas de tempo. Na escala de minutos e dias, os programadores escrevem testes de unidade para toda a lógica do sistema sujeita a problemas; através destes testes, os programadores têm, a cada minuto, uma realimentação concreta do estado do sistema. Na escala de semanas e meses, clientes e *testers*, membros da equipe, cuja principal responsabilidade são os testes funcionais, escrevem tais testes para todas as histórias, que são como casos de uso simplificados, implementadas no sistema. A cada duas ou três semanas, os clientes revisam o cronograma para verificar se a velocidade da equipe está compatível com o planejamento inicial ou se este deve ser ajustado (BECK, 2000).

Muitas vezes, as ações corajosas podem trazer um resultado positivo para um projeto. Eliminar um código que não funciona após um dia de trabalho e refazê-lo totalmente, em vez de tentar consertá-lo, ou pôr em prática idéias que, mesmo parecendo absurdas, possam reduzir a complexidade do sistema pode, causar grandes melhorias no projeto (BECK, 2000).

2.2 Princípios básicos do XP

Princípios, no contexto do XP, são as características essenciais de um projeto, baseadas nos valores da metodologia, promovidas pela equipe e implícitas nos produtos.

São princípios básicos do XP (BECK, 2000):

- A realimentação rápida: pois, de acordo com a psicologia, o tempo entre uma ação e sua realimentação é crítico para o aprendizado;
- A simplicidade assumida: todo problema deve ser tratado como se pudesse ser resolvido de modo extremamente simples;
- A mudança incremental: grandes mudanças feitas de uma única vez não funcionam;

- A adoção da mudança: a melhor estratégia é aquela que valoriza mais opções e realmente resolve os problemas prioritários;
- O trabalho de qualidade: pois todos gostam de fazer um bom trabalho.

Outros princípios, de menor importância, são (BECK, 2000, p.39):

- Ensinar a aprender: XP ensina a estimar quantos testes, refactoring e reestruturação do projeto devem ser feitos;
- Pequeno investimento inicial: um orçamento enxuto leva a se fazer um bom trabalho, pois resulta na priorização de atividades realmente importantes. Mas o investimento inicial deve ser suficiente para que se solucione, ao menos, um problema interessante;
- Jogar para vencer: a maioria dos projetos de software são feitos para não perder; preenchem-se documentos e são feitas reuniões, não porque estes façam sentido, mas para que no final, se houver algum erro, os culpados possam ser eximidos por estarem seguindo um processo. Jogar para vencer, ao contrário, significa que todas as medidas que possam levar ao sucesso do projeto serão tomadas, e todas as que não o fizerem não serão executadas.
- Experimentos concretos: todas as decisões abstratas devem ser testadas;
- Comunicação aberta e honesta: programadores devem ser capazes de explicar as consequências das decisões de outras pessoas, devem ser livres para expressar seus medos e pedir auxílio, e também devem se sentir livres para dar más notícias ao cliente e ao gerente;
- Trabalho a favor do instinto das pessoas: porque as pessoas gostam de ganhar, de aprender e de interagir umas com as outras;
- Aceitação de responsabilidades: responsabilidades devem ser aceitas, não dadas, porque "Nenhum procedimento é tão negativo para um projeto ou uma pessoa quanto a imposição de ordens" (BECK, 2000, p.41);
- Adaptação local: as normas do XP devem ser adaptadas para cada projeto;

- Viagem leve: os artefatos que cada membro da equipe devem manter consigo devem ser poucos, simples e de valor;
- Medição honesta: deve ser utilizada uma métrica correta e realista.

2.3 Divisão do trabalho no XP

O XP apresenta bons resultados se os membros da equipe de projeto desempenham adequadamente os seus papéis.

A seguir são descritos os papéis que membros de uma equipe XP podem assumir (BECK, 2000):

- Programador: aquele que produz o código, a peça fundamental de um projeto XP. Ele deve programar de forma simples e lógica, para garantir que os demais programadores entendam o código gerado.
- Cliente: Assim como o programador, é essencial para um projeto XP. De acordo com (BECK, 2000, p.142), "O programador sabe como programar. O cliente sabe o que programar"; um cliente XP deve saber como contar boas histórias, para tornar eficiente o trabalho dos programadores, deve saber tomar decisões, para atender prazos e definir prioridades e, de preferência, deve ser o usuário final do sistema que se está produzindo.
- Tester: responsável por, junto com o cliente, definir quais os testes funcionais e por repeti-los periodicamente.
- Tracker: responsável por acompanhar o andamento do trabalho de toda a equipe e estimar prazos, como a conclusão de iterações; "um bom *tracker* deve interrogar e informar a equipe sem amendrontá-la" (BECK, 2000, p. 144).
- Coach: é o responsável por todo o projeto; ajuda a manter o foco da equipe e a acalmá-la. O *coach* deve conhecer melhor que qualquer um na equipe as práticas do XP, para adequar a elas todas as etapas do projeto e instruir a equipe, não a confundindo.
- Consultor: responsável por suprir as deficiências técnicas da equipe XP. Usa seu conhecimento e experiência, quando necessário, para fazer parte do projeto.

- *Big Boss*: é um agente motivador para a equipe, que tem como função informá-la de qualquer desvio tão logo este ocorra para que haja mais tempo para resolvê-lo. Também deve reconciliar desentendimentos entre os membros da equipe.

Para que uma equipe funcione nos moldes do XP, é necessário que nela haja, pelo menos, os seguintes papéis: programador, cliente, *coach* e *tracker* (BECK, 2000).

2.4 Práticas do XP

Práticas, no contexto do XP, são ações planejadas e desenvolvidas pelos membros da equipe, motivadas pelos valores e regidas pelos princípios da metodologia, cuja execução conjunta representa o seguimento do XP.

As práticas do XP são (BECK, 2000):

- *Planning Game*: planejamento rápido que determina o escopo da próxima entrega, combinando prioridades do negócio e estimativas técnicas.
- Pequenas entregas: a cada iteração, cuja duração não deve ultrapassar um mês, deve ser entregue um sistema de software funcional ;
- Metáfora: guia o desenvolvimento como uma história simples e difundida de como o sistema funciona;
- *Design simples*: o sistema deve ter o *design* mais simples possível a todo momento;
- Testes: programadores continuamente desenvolvem testes de unidade que devem ser executados com sucesso para que o desenvolvimento continue. Clientes escrevem testes para verificar que as funcionalidades do sistema estão prontas;
- *Refactoring*: programadores reconstroem o sistema, sem alterar seu funcionamento, para eliminar duplicidades, simplificá-lo ou dar-lhe flexibilidade;
- Programação em pares: toda a codificação deve ser feita por dois programadores em uma única máquina;
- Propriedade coletiva: todos podem modificar qualquer parte do sistema a qualquer momento;

- Integração contínua: o sistema deve ser integrado várias vezes por dia, cada vez que uma tarefa é concluída;
- 40 horas por semana: não trabalhar mais que 40 horas por semana é uma regra;
- Clientes *on-site*: um usuário final do sistema fica sempre disponível para responder a questões;
- Padrões de codificação: os programadores devem escrever o código de acordo com regras, enfatizando o código como meio de comunicação.

2.5 Ciclo de vida de um projeto XP ideal

O ciclo de vida de um projeto XP ideal é dividido em 6 fases (BECK, 2000):

- Exploração: deve ser tão curta quanto possível; é uma fase de pré-produção que estará concluída quando o cliente considerar que há material suficiente nos cartões de histórias para o desenvolvimento de uma primeira versão e os programadores não puderem estimar mais sem efetivamente implementarem o sistema.
- Planejamento: é a fase em que o cliente e os programadores chegam a um acordo quanto ao menor número de histórias que estarão implementadas na data limite. Dura um ou dois dias.
- Iterações para primeira edição: o cronograma é dividido em iterações, cujo resultado são casos de teste aplicáveis a cada uma das histórias. Durante a primeira iteração o esqueleto de todo o sistema deve ser criado. Cada iteração deve durar entre uma e quatro semanas.
- Preparação para produção (*productionizing*): é a etapa que engloba as últimas iterações que antecedem a colocação do sistema em produção. Nesta etapa as iterações devem ser mais curtas (não mais que uma semana) e devem ser realizadas reuniões diárias para que toda a equipe conheça o trabalho de cada um.
- Manutenção: etapa em que, simultaneamente, são produzidas novas funcionalidades, o sistema é mantido em funcionamento, são inseridas novas pessoas à equipe e algumas mudam de posto.

- **Morte:** é a fase em que o cliente não consegue imaginar novas histórias para acrescentar ao projeto ou a equipe de desenvolvimento é incapaz de agregar ao sistema as funcionalidades desejadas pelo cliente. Nesta etapa decide-se qual a melhor alternativa para substituição do sistema em uso e durante ela é feita apenas a manutenção corretiva, até que o sistema seja efetivamente substituído.

3 Descrição do sistema

O sistema desenvolvido tem por função agilizar as consultas que médicos fazem aos dados de seus pacientes antes de ministrar os anestésicos; visa a substituir formulários em papel, de uso corrente, associando a portabilidade de tais formulários com a segurança e a organização dos computadores móveis (*Pocket PC*), que serão os instrumentos usados para tais consultas. O sistema conta ainda com um módulo, através do qual médicos e enfermeiras também podem cadastrar e consultar dados clínicos dos pacientes.

A figura a seguir representa a arquitetura do sistema; o módulo web é identificado pela sigla SPW (Sistema Pré-anestésico Web) e o módulo executado nos dispositivos móveis é identificado como SPM (Sistema Pré-anestésico Móvel).

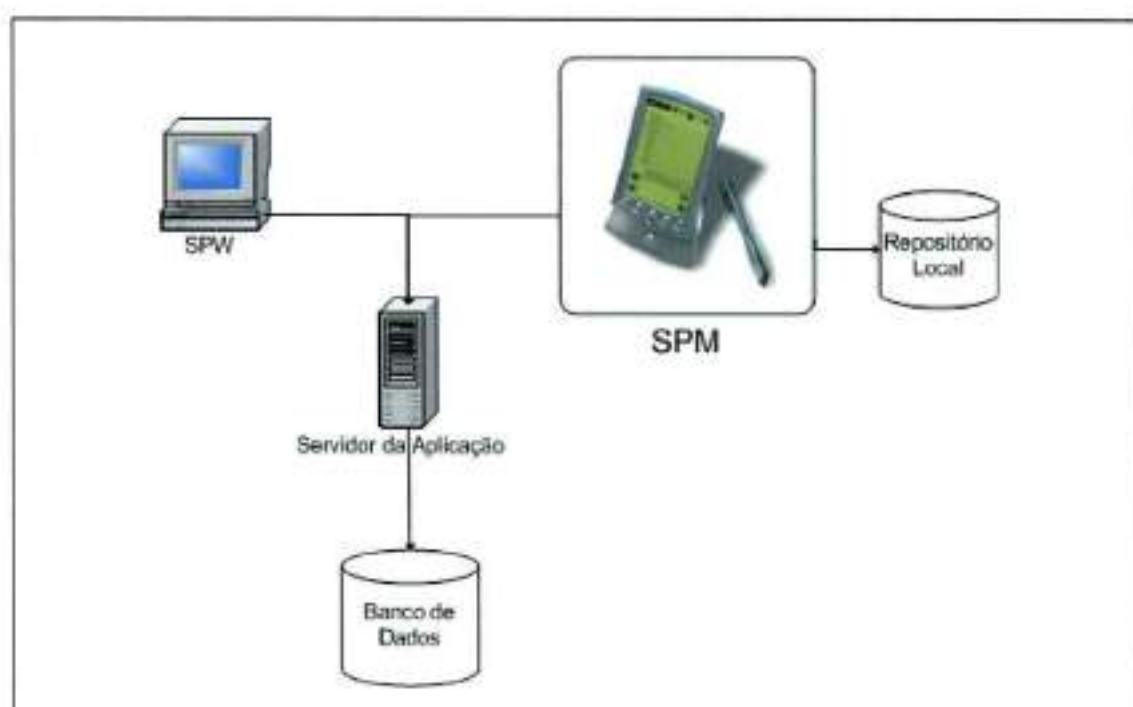


Figura 1: Arquitetura do sistema

O SPM tem um repositório local para armazenamento exclusivo dos dados de pacientes escolhidos pelo médico/usuário. O servidor de aplicação faz a serialização e criptografia de dados. Mais detalhes técnicos do sistema estão nos documentos de especificação de requisitos de software (Anexos 8.1 e 8.2).

Serão armazenadas, no sistema, informações cadastrais dos pacientes, informações de acesso de médicos e fichas de anamnese dos pacientes. Tais dados são, respectivamente, cadeias de caracteres, números inteiros e valores booleanos.

Quanto à utilização, os dados armazenados no sistema podem ser divididos em:

- Dados de uso frequente: devem estar sempre disponíveis no banco de dados principal. São eles:
 - Nome de usuário e senha para *log in*;
 - Informações cadastrais e fichas clínicas de pacientes que mantêm um acompanhamento constante no laboratório;
- Dados de uso esporádico: devem ser mantidos no banco de dados principal pelo período entre duas visitas ao laboratório, que é tipicamente de 6 meses, e depois incluídos no banco de dados de histórico. São eles:
 - Informações cadastrais e fichas clínicas de pacientes que buscam o laboratório apenas em intervalos longos, de anos, para obtenção de diagnósticos ou tratamentos específicos;
- Dado de uso único: devem ser armazenados como os dados de uso esporádico. São eles:
 - Laudos úteis para realização de cirurgia, que uma vez feita, torna desnecessários posteriores diagnósticos e tratamentos;
 - Dados de pacientes que trocam de laboratório.

Cada alteração de dados no sistema é precedida do arquivamento, em um histórico, dos dados originais, da data e do responsável pela alteração. Isto evita que informações importantes sejam perdidas e torna possível acompanhar toda a evolução de um paciente, desde seu primeiro atendimento no laboratório, sem que a base de dados principal seja sobrecarregada.

4 Aplicação do XP

Para o desenvolvimento do sistema descrito no capítulo anterior, foram aplicados os valores, princípios e práticas do *Extreme Programming*. Este capítulo descreve as circunstâncias de aplicação e como o trabalho e a equipe foram organizados para a implantação da metodologia.

4.1 Processo Aplicado

O XP não é descrito como uma sequência de atividades, com resultados definidos, que se combinam numa sequência definida para a obtenção de um produto final; ele é descrito como uma série de práticas que devem ser executadas para gerar produtos com características definidas. Para que garantir a aplicação organizada de tais práticas neste trabalho, elas foram concatenadas, o que resultou no processo descrito nesta seção.

A sequência de atividades realizadas durante a aplicação do XP neste trabalho é apresentada na figura 2.

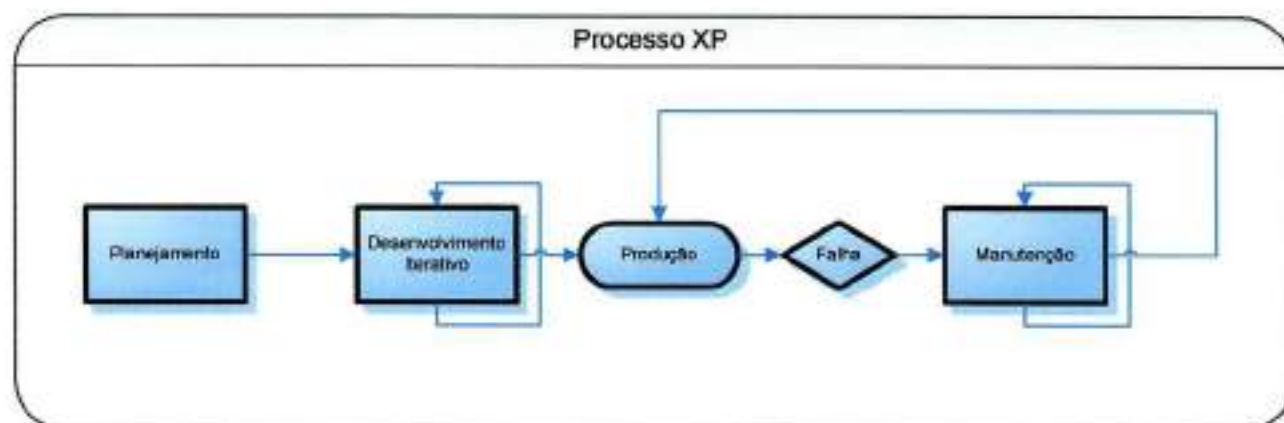


Figura 2: Processo aplicado

Este processo é baseado no ciclo de vida de um projeto XP ideal. A fase de planejamento deste processo inclui a de exploração, do ciclo de vida de um projeto XP ideal, e nela desenvolvem-se *story cards* casos de testes (funcionais) e o cronograma de iterações (resultante do *Planning Game*); as fases de exploração e planejamento foram unidas porque ambas tem como objetivo planejar o projeto, seja quanto ao seus requisitos técnicos, como é o caso da exploração, ou da sua distribuição cronológica, como é o caso da fase de planejamento. O desenvolvimento e a manutenção, são feitos em iterações e seguem o padrão detalhado na figura 3; nestas etapas são realizadas sessões de programação em que são implementadas novas funcionalidades do sistema; quando todas as funcionalidades previstas para uma iteração estiverem prontas, parte-se para o *productionizing* em que são feitos os ajustes necessários para a entrega do sistema.

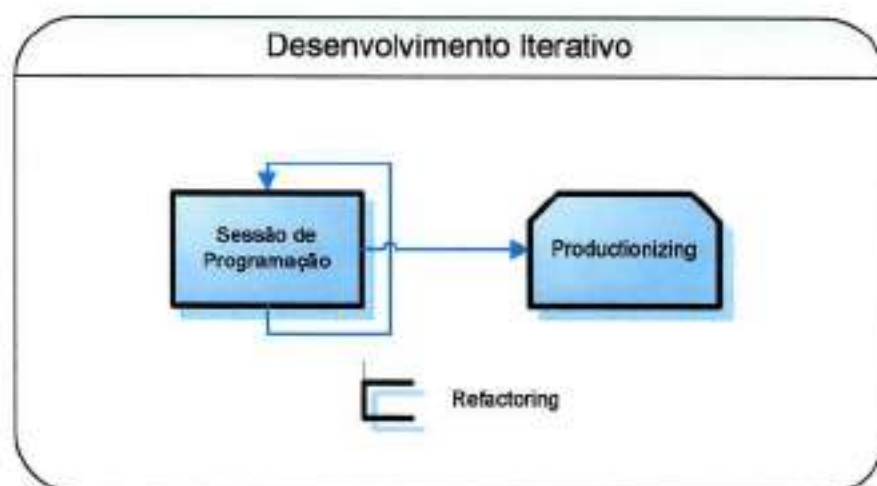


Figura 3: Detalhamento da atividade iterativa

Em cada etapa do processo, obtiveram-se resultados específicos, conforme ilustrado nas figuras 4,5 e 6.

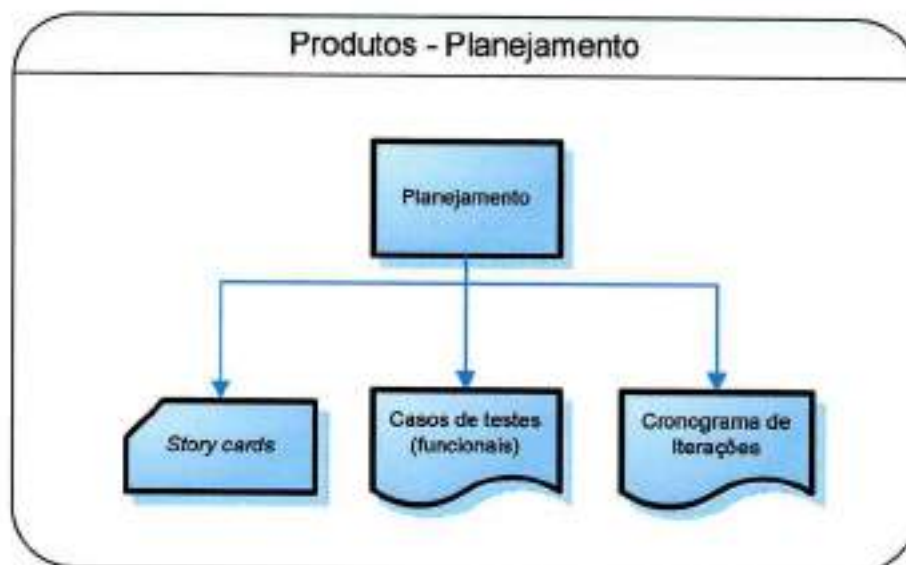


Figura 4: Produtos resultantes do planejamento

A etapa de planejamento desenvolveu-se através das reuniões envolvendo o cliente e o corpo técnico da equipe, quando foram produzidos *story cards* concisos ou foram aprovados os casos de testes funcionais, escritos pela equipe técnica. Com base nas especificações do sistema, definidas nesta etapa, foi produzido o cronograma inicial de iterações.



Figura 5: Produtos resultantes das sessões de programação

A cada sessão de programação, um módulo funcional foi agregado ao sistema, ou seja, uma nova versão do sistema foi produzida. Para testar cada módulo ou alguns trechos de código, os programadores escreveram testes. Para gerenciamento cronológico das sessões, o *coach* produziu um *task card* para cada sessão de programação realizada.

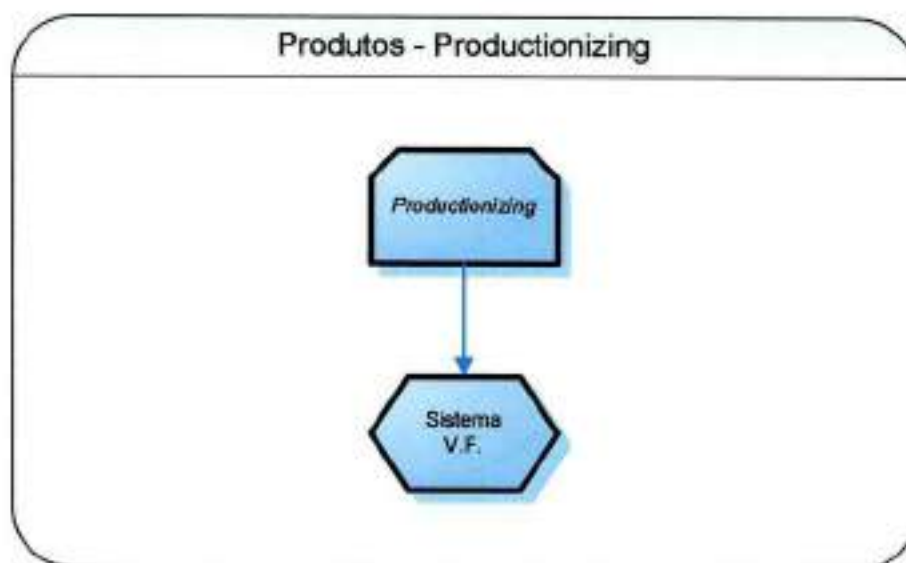


Figura 6: Produtos resultantes do *productionizing*

A fase de *productionizing*, ou seja, a última sessão de programação antes da entrega do sistema, tem como produto a versão do sistema prometida para cada iteração.

4.2 Divisão de trabalho na equipe

Para se adequar ao XP, a equipe foi constituída por dois programadores, um *coach*, que também teve o papel de *tracker*, e dois clientes. Para que os prazos especificados no planejamento fossem respeitados, também houve um consultor, que integrou a equipe em 8 das 54 tarefas em que o desenvolvimento foi dividido.

Os programadores alternaram-se nos papéis principal, aquele que realmente gera código, e secundário, aquele que revisa o código gerado pelo primeiro em tempo real e ainda sugere mudanças estruturais e o guia ao uso constante de padrões. O consultor, que teve como tarefa principal produzir interfaces homem-computador, atuou inicialmente no papel principal das sessões de programação, sempre acompanhado de um programador como revisor e aprendiz e, por fim, passou para a função secundária, para solucionar as dúvidas remanescentes do programador principal.

Os clientes foram um representante do laboratório de análises médicas a que o sistema se destina e um assessor. Estes, devido ao seu envolvimento limitado com o projeto, precisaram de apoio do resto da equipe para definição funcional do sistema e para escolha das prioridades de cada iteração. Ainda assim, os clientes foram os responsáveis por todas as decisões não técnicas do sistema.

O *coach* teve como tarefas principais determinar quais os passos a serem tomados para que a equipe seguisse as práticas previstas pelo XP e, ao assumir a posição de *tracker*, verificou o rendimento da equipe e tomou as medidas necessárias para que fossem respeitadas as datas de término de cada iteração.

A equipe foi segmentada em uma parte técnica, formada pelos programadores e o consultor, que estiveram relacionados diretamente à programação, e uma parte gerencial, formada pelo *coach* e os clientes.

4.3 Participação do cliente

Os clientes não participaram tão ativamente do projeto como prevê a literatura (BECK, 2000). Eles não elaboraram os *story cards*, nem os testes funcionais. Os *story cards* foram resumidamente definidos nas reuniões iniciais do projeto e também naquelas que antecederam a segunda iteração, quando o desenvolvimento do módulo *web* foi incluído no escopo do projeto; o detalhamento foi feito pela equipe técnica. Os testes funcionais

foram descritos pelos programadores e pelo *coach* em conjunto e posteriormente revisados e aprovados pelo cliente.

No entanto, todas as dúvidas sobre as funcionalidades do sistema que os programadores tiveram ao longo do desenvolvimento foram comunicadas ao cliente rapidamente; reuniões diárias foram realizadas com este propósito durante as iterações para que, além de orientar os programadores, o cliente pudesse verificar constantemente o progresso do desenvolvimento.

4.4 Story Cards e Task Cards

Durante as reuniões em que foram elaborados *story cards*, que ocorreram no início do projeto e antes da segunda iteração, quando se incluiu no escopo o módulo web (SPW), foram registradas as descrições resumidas e a prioridade de cada tarefa. Posteriormente, os programadores e o *coach* escreveram, em conjunto, os testes funcionais para cada *story card* com prioridade alta, fizeram estimativas de tempo necessário para implementar cada uma destas e sugeriram reduções de escopo quando houve incompatibilidade entre o tempo estimado e o disponível para uma iteração. Após a aprovação do escopo e dos testes funcionais pelo cliente, o *coach* desenvolveu um *task cards* para cada *story card*.

Cada *task card* é composto das tarefas de programação que, em conjunto, têm como produto uma funcionalidade do sistema, ou seja, uma história do cliente. Cada uma destas tarefas é apresentada com uma breve descrição, uma estimativa de horas necessárias para realizá-las, feita pelo *coach* com apoio dos programadores, o número de horas efetivamente gastas para realizá-la, o programador principal e o secundário que participaram da *sessão de programação* de execução da tarefa.

4.5 *Planning Game (iterações e prioridades)*

Nas reuniões de definição dos *story cards*, os clientes definiram, inicialmente, todas as funcionalidades que desejavam para o sistema. Logo após isto, precisaram escolher, entre as funcionalidades definidas, quais as que tinham prioridade alta. O *Planning Game*, que é o processo de escolha das prioridades e divisão do desenvolvimento em iterações, que devem terminar na data prevista para colocação do sistema final em produção, começou com a divisão do projeto em duas iterações, com as seguintes prioridades:

- Primeira iteração: desenvolvimento das funcionalidades principais do módulo móvel e integração deste à arquitetura existente;
- Segunda iteração: desenvolvimento das funcionalidades não tratadas na primeira iteração para o módulo móvel e integração.

Depois disso, o coach verificou cada *story card*, através do *test case* associado a ela, o qual, além de guiar a validação de uma funcionalidade pronta, serve para avaliar sua complexidade e dependências, e da estimativa de tempo necessário para desenvolvê-la.

Após a primeira iteração o cliente incluiu o desenvolvimento do módulo *web* no escopo do projeto, porque tal módulo, que já havia sido desenvolvido anteriormente por uma outra equipe, não estava funcionando e seus desenvolvedores não estavam aptos a fazer a manutenção necessária. Após um novo *Planning Game* o desenvolvimento foi dividido em três iterações. A primeira delas, quando o sistema final desejado pelo cliente limitava-se ao módulo móvel (SPM), teve todos os *story cards* de alta prioridade, escolhidos em reunião com o cliente, aceitos e implementados. Na segunda iteração, o cliente deu prioridade alta ao desenvolvimento de todo o módulo *web* (SPW); após avaliar os *test cases* e estimativas, o coach pediu ao cliente que escolhesse as funcionalidades de menor prioridade, para que estas fossem realizadas na terceira iteração. Escolhidas estas funcionalidades, definiu-se toda a terceira iteração, que também seria de adaptação das histórias realizadas antes da inclusão do SPW ao escopo do sistema.

4.6 Sessões de programação

As sessões de programação foram os eventos de programação em pares realizados para produção de todo o código gerado. Como exigiam a presença simultânea de dois programadores, tais eventos tiveram de ser agendados. Duas sessões de programação foram feitas como conferências, utilizando-se sistemas de comunicação instantânea pela internet (*Windows Messenger* ou *Skype*) e troca de código por email.

Para as sessões de programação, que foram feitas com a presença de dois programadores, utilizaram-se diversas variantes do modelo descrito em (BECK, 2000), com dois programadores usando o mesmo computador. Até o final da segunda iteração, 10 sessões de programação foram feitas desta maneira, mas, para ganho de eficiência, houve duas sessões em que se utilizaram dois monitores com imagem replicada, e uma com dois computadores para que as dúvidas do programadores pudessem ser encontradas mais rapidamente em tutoriais ou sites de programação na internet.

4.7 Testes

Os testes funcionais foram colocados no mesmo banco de dados dos *story cards*, assim puderam ser validados pelo cliente e posteriormente executados para avaliação de cada *story card*.

Os testes de unidade foram listados pelos programadores e executados a cada integração numa iteração.

4.8 Comunicação

A comunicação entre os membros da equipe ocorreu em reuniões rápidas, que se realizaram diariamente durante o andamento de cada iteração; estas reuniões foram realizadas para que toda a equipe tomasse conhecimento das dificuldades e rendimento

do trabalho. Foram também realizadas reuniões agendadas, de maior duração, para definição de *story cards*, de que participaram, pelo menos, um cliente, o *coach* e um dos programadores.

Como a comunicação é um dos valores do XP, todas as dúvidas sobre funcionalidades do sistema foram sempre comunicadas ao cliente, assim como mudanças de cronograma que se tornaram necessárias ao longo do desenvolvimento.

4.9 Padrões de programação

Foram estabelecidos, entre os programadores, os seguintes padrões de programação:

- O acesso aos dados do sistema foi feito apenas através de web services, para que se garantisse a semelhança estrutural entre seus módulos;
- Utilização de tipos das variáveis em sua nomenclatura; assim, uma variável de tipo inteiro (int), por exemplo, deve ser nomeada como intXXX.

4.10 Sistema de apoio ao projeto

O sistema de apoio ao projeto foi desenvolvido para garantir que as práticas previstas na literatura para o XP fossem usadas de modo eficiente; tal sistema é composto por um banco de dados de *story cards*, casos de teste e *task cards*, além de uma interface capaz de mostrar juntamente dados correlatos, isto é, casos de teste e *task cards* são apresentados juntamente com o *story card* a eles relacionado. Além disso, o sistema conta com uma inteligência para cálculo de horas trabalhadas e estimadas para as tarefas finalizadas e pendentes.

O conteúdo de cada *story card* foi baseado num modelo presente em (BECK, 2000); além de uma descrição resumida da funcionalidade a ser desenvolvida, cada *story card* tem a data em que foi inserido no banco, o tipo de atividade, se nova, de ajuste ou ampliação, a prioridade, se alta, baixa ou média, o cliente que participou da definição da estória, o

risco, que depende da complexidade da funcionalidade e do impacto que ela tem sobre as demais partes do sistema, e uma estimativa inicial do tempo necessário para a implementação da funcionalidade, feita pela equipe técnica.

São apresentadas, nas figuras 7 e 8, as telas disponíveis no sistema de apoio ao projeto.

(2)

(1)

(4)

(3)

(5)

Data	Status	Pendências	Comentários
13/8/2005	pendente		aprovada pelo cliente para primeira iteração
24/8/2005	finalizada		
26/9/2005	pendente	adaptação - novo SPW	prevista para terceira iteração

Figura 7: Sistema de apoio ao projeto – tela principal

Na tela principal (figura 7), durante as reuniões dedicadas à definição dos *story cards*, foram inseridas uma descrição resumida de cada um destes no campo indicado pelo balão (1), além das atribuições formais do *story card*, nos campos indicados por (2). Após as reuniões, foram desenvolvidos os casos de teste funcionais, inseridos na página *Test Case*, indicada por (3). Na página *Andamento*, designada pelo balão (4), foram inseridos todos os eventos importantes relacionados a cada *story card*, como sua aprovação pelo cliente, finalização ou necessidade de manutenção. A página *Task Card*, indicada por (5), traz todas as sub-tarefas necessárias para a execução de um *story card*, seu status,

estimativa de tempo para sua realização e, no caso de tarefas finalizadas, os programadores responsáveis e o tempo real de trabalho.

Durante o desenvolvimento, o sistema de apoio ficou disponível a todos os membros da equipe para consulta e alteração.

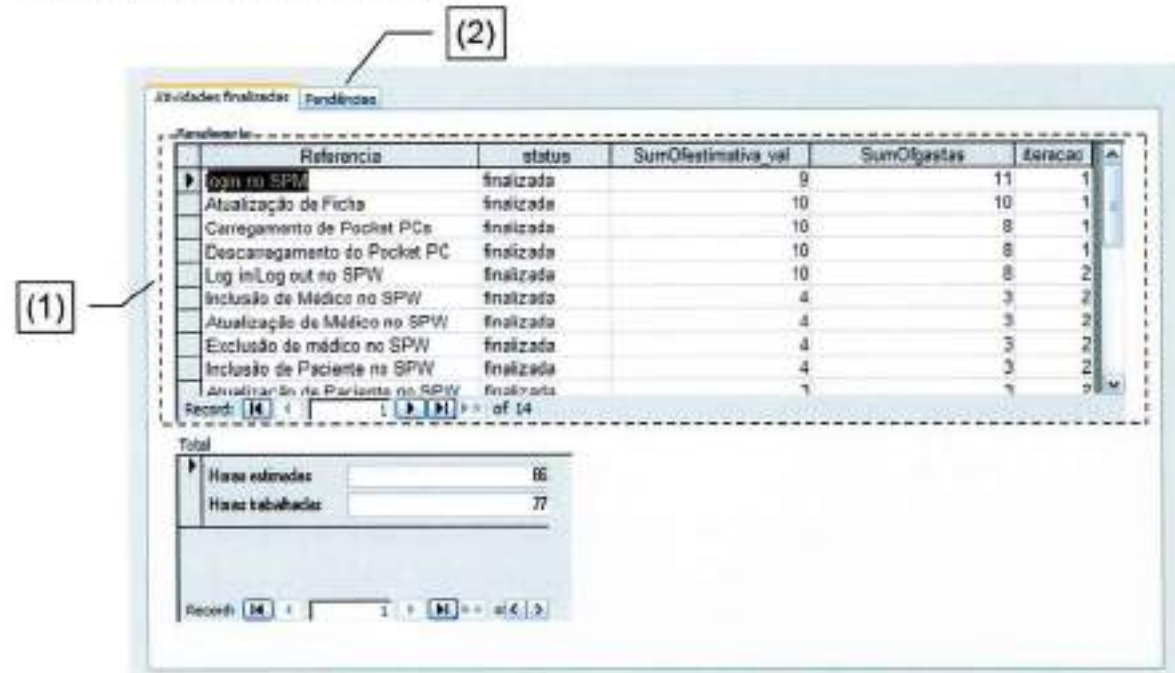


Figura 8: Sistema de apoio ao projeto – tela de rendimento

Na tela de rendimento (figura 8), é possível verificar, para cada história implementada, as horas estimadas para sua conclusão, o tempo real gasto para tal fim e a iteração em que ela foi desenvolvida, conforme apresentado na tabela indicada pelo balão (1). Também é possível verificar o total de horas estimadas e realmente gastas para conclusão das histórias implementadas.

Na página indicada pelo balão (2) são apresentados dados análogos, referentes às estórias pendentes. Neste caso não é mostrado o número de horas gastas.

5 Avaliação da aplicação

Neste capítulo é feita uma avaliação da aplicação das práticas e valores do XP no planejamento e desenvolvimento do sistema apresentado anteriormente.

5.1 Equipe

A equipe foi integrada por dois clientes que não se dedicaram exclusivamente ao projeto; apesar da orientação presente em (BECK, 2000) para que o cliente fique no mesmo espaço físico que os programadores, e neste local escreva *story cards* e testes funcionais, além de executarem os testes que escreveram. Os clientes continuaram envolvidos com outros projetos e atuaram apenas na validação de *story cards*, testes e de cada sub-produto, interagindo com os demais membros da equipe em reuniões.

Para que os clientes se envolvessem com as atividades que deveriam executar, os *story cards* e os testes funcionais foram escritos pelo *coach* e os programadores, e submetidos à validação daqueles. Esta participação assistida solucionou a dificuldade de se manter um cliente *in house*, já que, como resultado dela, obtiveram-se *story cards* e testes funcionais cujo responsável foi o cliente.

Para que os *story cards* e os testes funcionais fossem compatíveis com as necessidades do cliente, foram cadastradas, em reuniões marcadas com esta finalidade, descrições resumidas de cada *story card* no sistema de apoio ao projeto XP. A descrição deste sistema se encontra na seção 4.10. Os testes funcionais também foram cadastrados neste sistema, mas depois das reuniões, devido ao seu nível de detalhamento e à sua extensão. Deste modo o cliente ganhou o papel de co-autor das histórias do sistema.

Os membros da equipe, exceto o cliente, tiveram tarefas bem definidas e conseguiram executá-las sem que fossem necessárias sobreposições de incumbências; os

programadores apenas dedicaram-se à estruturação técnica e implementação do sistema, e compartilharam entre si informações para resolução conjunta das dificuldades contidas em tais tarefas; o *coach* planejou os trabalhos, com a ajuda do sistema de apoio ao projeto, orientou a equipe para uso constante das práticas do XP e, como *tracker*, avaliou o andamento do projeto. As atividades que não puderam ser feitas pelo cliente, a edição de *story cards* e os casos de teste, foram realizadas, em conjunto, pelos demais membros da equipe. Portanto, a equipe em questão, com os papéis básicos do *Extreme Programming*, foi capaz de planejar e desenvolver um projeto de software, e ainda arcar com a deficiência de um de seus integrantes.

5.2 Escolha de prioridades

A escolha de prioridades causou, no início do projeto, desconfiança no cliente quanto ao potencial da equipe. Antes da segunda iteração, o escopo foi reduzido contra a vontade do cliente, porque o tempo estimado pela equipe para implementação de todas as funcionalidades com prioridade alta era superior ao disponível, devido a isso, a avaliação que o cliente fez da equipe foi negativa. Esta redução do escopo, ainda que tenha minimizado o risco de exceder o prazo de entrega da iteração, também impossibilitou o desenvolvimento de um produto de maior valor agregado, dado o maior número de funcionalidades de que ele disporia na versão original. No entanto, ao final da segunda iteração, verificou-se que a estimativa de tempo inicial foi exagerada, e que o escopo inicial poderia ser implementado.

Nenhuma das iterações excedeu o prazo limite; ainda que os produtos entregues ao final de cada uma delas não tenha sido exatamente o que o cliente desejava, foi o prometido, e isto deu confiança ao cliente.

5.3 Iterações

A divisão do desenvolvimento em iterações foi importante para a sobrevivência do projeto. Ao final da primeira iteração, o cliente verificou que tinha dois produtos entregues, o módulo *web* (SPW – Sistema Pré-anestésico *Web*) deficiente (desenvolvido por um outro grupo) e o módulo móvel (SPM – Sistema Pré-Anestésico Móvel), produto da iteração com XP, funcional, testado e validado por ele próprio. A deficiência do módulo *web* e a confiança que o cliente depositava no módulo móvel, dada a proximidade com que acompanhou seu desenvolvimento, foram definitivas para que o projeto não fosse abandonado, e o módulo *web* fosse reprojetoado pela equipe do XP.

Desenvolver em iterações e, através delas, gerar sub-produtos funcionais em intervalos curtos, dá confiança ao projeto e auxilia a cobrança pelo trabalho realizado; a cada sub-produto entregue, uma fração do valor tratado para o projeto deve ser paga pelo cliente. Se o projeto for abandonado, as perdas são parciais tanto para a equipe técnica, que recebeu pelos sub-produtos entregues, quanto para o cliente, que dispõe de tais sub-produtos funcionais.

Além disso, o desenvolvimento da primeira iteração ocorre num intervalo de tempo similar ao necessário para desenvolvimento de um protótipo do sistema completo e, assim como este, o sub-produto da primeira iteração é capaz de conquistar ou não a confiança do cliente.

5.4 Programação em pares

A cada sessão de programação, os programadores listaram as vantagens e desvantagens de se trabalhar em duplas, e ainda fizeram sugestões para que este modo de trabalho fosse melhorado. Destes relatórios foram extraídas as constatações descritas a seguir.

A principal vantagem obtida no desenvolvimento com duplas de programadores foi a correção rápida de erros; o programador secundário é capaz de aplicar melhor que o principal as regras da linguagem de programação, já que não está tão preocupado como primeiro com a funcionalidade que o código deve gerar. O programador secundário não só antecipa o trabalho do compilador, de verificar erros sintáticos, mas também inibe a reprodução de código errado.

O programador secundário estimula a aderência aos padrões de programação pré-estabelecidos e, portanto, promove o caráter coletivo do código gerado com a metodologia XP.

A troca de informações entre o programador principal e o secundário ajuda a resolver dúvidas que o primeiro tem a respeito da linguagem de programação, das funcionalidades da ferramenta de desenvolvimento adotada e ainda promove a simplicidade do código, uma vez que o programador secundário pode sugerir o uso de recursos, desconhecidos pelo principal, capazes de implementar a mesma função que este implementaria de modo complexo.

A principal desvantagem da programação em pares foi a necessidade dos programadores do par estarem juntos para que ocorresse. Para diminuir a possibilidade de cancelamento de sessões de programação, estas foram sempre agendadas com antecedência; porém, mesmo com este cuidado, houve cancelamentos ocasionados pela falta de um dos programadores, o que causou desperdício de recursos e atrasos que não aconteceriam num projeto convencional, em que a programação seria feita individualmente.

No projeto desenvolvido, os programadores eram bem entrosados e isto beneficiou muito a produtividade das sessões. Ainda assim, os programadores sugeriram que houvesse maior independência entre eles, com a utilização de um monitor para o programador principal e outro, replicado, para o secundário; assim ambos poderiam visualizar o código

mais confortavelmente. Também sugeriram que o programador secundário tivesse seu próprio computador, para procurar na internet soluções desconhecidas pela dupla.

5.5 Testes funcionais

Os testes funcionais constituem a descrição mais precisa do produto final e sub-produtos de iterações, a qual deve ser aprovada pelo cliente e com a qual contam tanto a equipe técnica como o próprio cliente como garantia. Os testes funcionais foram escritos em linguagem natural como uma sucessão de comandos e resultados obtidos através destes. Um teste funcional é executado com sucesso num sistema de software quando todos os comandos previstos no teste são executados neste sistema e ele retorna todas as respostas previstas no teste. Validar o sistema significa executar com sucesso todos os testes funcionais escritos.

Uma medida tomada no projeto para facilitar a execução dos testes funcionais, e consequentemente a validação do sistema, foi armazená-los no sistema de apoio ao projeto. Assim, o cliente pôde usar a mesma ferramenta para validar os testes e para guiá-lo na validação do sistema.

5.6 Reuniões

As reuniões diárias que ocorreram durante as iterações do projeto foram eficazes em transmitir aos membros da equipe a realimentação esperada. Como previsto em (BECK, 2000), as reuniões foram curtas, e cinco delas foram realizadas em conferências *on line*, e, por isso, foram eficientes.

As reuniões permitiram aos programadores comunicar o andamento do projeto ao *coach* e a este trouxeram a realimentação necessária para fazer novas estimativas e mostraram quando era necessário o trabalho de um consultor. Trouxeram à equipe técnica a opinião do cliente e a este a confiança de que o projeto seguia.

Das reuniões realizadas antes de cada iteração, para que fossem criados e aprovados os *story cards*, foram extraídas as informações que guiaram todo o projeto.

5.7 Padrões de programação

O uso de padrões promoveu a propriedade coletiva do código e o tornou mais simples e modular.

Ainda que os programadores tenham relutado em abandonar suas particularidades e vícios de programação, estes avaliaram positivamente os resultados da padronização do código, principalmente porque deixaram de ser os únicos responsáveis pelo código que produziram, uma vez que qualquer programador da equipe tornou-se capaz de fazer manutenção em qualquer código.

6 Considerações finais

Este capítulo descreve a avaliação do XP, realizada através de sua aplicação num projeto de software, suas contribuições à engenharia de software e aos processos de desenvolvimento de software, e os trabalhos futuros.

6.1 Conclusão

A principal vantagem constatada com a aplicação do XP foi a objetividade com que o projeto se desenvolveu; tendo como documentos oficiais apenas os *story cards* e casos de testes, a fase de planejamento pôde ser feita em três dias. A definição da arquitetura do sistema e o modelo do banco de dados foram desenvolvidos como documentos complementares, porque estes foram realmente necessários para guiar a produção do sistema.

A equipe interage num projeto XP durante as sessões de programação, nas reuniões de acompanhamento e de planejamento; durante as iterações, os programadores trabalham sempre juntos, e o *coach* deve acompanhar o trabalho diariamente, junto com estes. O cliente pode opinar a qualquer momento e começa a validar o sistema durante a fase inicial, de planejamento, quando deve aprovar, ou escrever, os testes funcionais. Quando uma equipe absorve as mudanças culturais do XP, o retorno é muito positivo: a propriedade do software produzido torna-se coletiva, padrões de programação são difundidos e aproveita-se sempre a especialização de cada membro da equipe na solução e otimização do sistema produzido.

Com a execução de todos os testes de unidade escritos a cada integração de um módulo ao sistema, reduz-se a ocorrência de erros oriundos de incompatibilidade. Durante as iterações desenvolvidas neste trabalho, o número de erros decorrentes de

incompatibilidades diminuía à medida que mais módulos eram integrados ao sistema, já que, com a execução de testes, os programadores tomavam conhecimento dos defeitos prévios do sistema e não os repetiam nos módulos consecutivos.

O resultado de cada iteração é um software realmente útil e, conseqüentemente, de maior valor agregado para o cliente; isto acontece porque são estabelecidas prioridades para cada iteração e, então, as funcionalidades principais são desenvolvidas primeiro. Estabelecer prioridades é uma maneira de se melhorar a avaliação que o cliente faz do sistema de software produzido, sem sobrecarregar a equipe técnica.

Atualmente, uma equipe ou uma empresa que deseje fazer um projeto usando XP tem que criar seu próprio processo, adaptando práticas discutidas em livros, congressos ou associações dedicadas às metodologias ágeis. Também não há certificação para entidades que usam XP; logo a metodologia acrescenta valor ao produto desenvolvido pelas qualidades deste, mas não por seu modo de desenvolvimento.

Uma outra desvantagem do XP é que este impõe mudanças culturais severas, que não são recebidas com muito agrado. Verificou-se neste trabalho que, até que os programadores não se sentissem beneficiados pela propriedade coletiva do código e pela divisão de responsabilidades sem ponderação por especialização, estes não viam as sessões de programação com bons olhos. O cliente não tinha motivação para participar das reuniões de acompanhamento até que, como resultado da primeira iteração, viu suas sugestões implementadas sem que fosse necessário encaminhar o sistema à manutenção. Porém, após a primeira iteração, os programadores estavam integrados e formavam um time colaborativo; os clientes, após receberem o primeiro sub-produto do sistema, totalmente compatível com suas pretensões, já que estas foram constantemente apresentadas nas reuniões, ganharam motivação.

6.2 Contribuições

Este trabalho contribui com o estudo dos processos para desenvolvimento de software na medida que teve, como resultados, um processo aplicado ao XP, que pode ser usado como base para qualquer projeto, uma ferramenta de acompanhamento para este processo, e métricas resultantes da avaliação, a cada iteração, de um projeto real. As métricas obtidas estão no relatório de produtividade em anexo na sessão 8.3.

Os resultados obtidos neste trabalho podem ser utilizados em outros projetos de software de pequeno ou médio porte como forma de se evitarem as deficiências do XP e de se aproveitarem suas vantagens.

6.3 Trabalhos futuros

Este trabalho pode ser continuado com o desenvolvimento de processos com base no XP aplicados a sistemas de software com complexidade e uso diversos; desta maneira haveria métricas mais precisas para cada tipo de projeto, e poderiam ser criadas ferramentas de acompanhamento mais apropriadas para cada um deles.

Criar um método de avaliação de uso do XP, para garantir que um projeto é desenvolvido com a metodologia, é também uma possível extensão deste trabalho. Desta maneira, empresas que usem a metodologia poderão ser certificadas e ter o valor de seus sistemas de software aumentado.

7 Bibliografia

ABES - Associação Brasileira das Empresas de Software; **Filiação à ABES**; Disponível em <<http://www.abes.org.br/templ1.aspx?id=3&sub=3>>; Acessado em 9/2005

AGILE MANIFESTO; **Authors: The Agile Manifesto**; Divulgado em 2001; Disponível em <www.agilemanifesto.org/authors.html>; Acessado em: 9/2005

BECK, K. **Extreme Programming explained: embrace change**. [S.1.]: Addison-Wesley Longman Publishing Co., Inc., 2000. ISBN 0-201-61641-6

BOVESPA – Bolsa de Valores do Estado de São Paulo; **Classificação setorial das empresas e fundos negociados na Bovespa**; Disponível em <<http://www.bovespa.com.br/pdf/ClassifSetorial.pdf>>; Acessado em: 9/2005

FAYAD, M. E.; LAITINEN, M.; WARD, R. P. Thinking objectively: software engineering in the small. **Communications of the ACM**, ACM Press, v. 43, n. 3, p. 114-118, 2000. ISSN 0001-0782.

HIGHSMITH, J. A. **Agile software development ecosystems** [S.1.]: Pearson Education, Inc., 2002. ISBN 0-201-76043-6

LAUDON, K. C.; LAUDON, J. P.; BRABSTON, M. E. **Management of information systems: managing the digital firm**. [S.1.]: Pearson Education Canada, Inc.; Prentice Hall, Inc. 2002. ISBN 0-13-033809-5

NASDAQ - *National Association of Securities Dealers Automated Quotation System*; **Sectoring by industry groups** Disponível em: <www.nasdaq.com/reference/BarChartSectors.stm>; Acessado em 9/2005

WARD, R. P.; FAYAD, M. E.; LAITINEN, M. Thinking objectively: software process improvement in the small. **Communications of the ACM**, ACM Press, v. 44, n. 4, p 105-107, 2001. ISSN 0001-0782.

SHARMA, PRADYUMN; **An introduction to Extreme Programming**; Divulgado em: 2004; Disponível em: <www.advisor.com/doc/13571>; Acessado em: 9/2005.

STEPHENS, M; ROSENBERG, D. **Extreme Programming refactored: the case against XP**. [S.1.]: Apress; Springer, 2003. ISBN 1-59059-096-1

8 Anexos

Estão em anexo os seguintes documentos:

8.1 Especificação de Requisitos de Software – SPM

Este documento apresenta uma descrição técnica do SPM (Sistema Pré-anestésico Móvel); esta descrição é feita através de story cards, testes funcionais e informações sobre a arquitetura em que o SPM se insere, suas dependências e restrições.

8.2 Especificação de Requisitos de Software – SPW

Este documento é complementar ao do item 8.1; ele apresenta uma descrição do SPW (Sistema Pré-anestésico Web).

8.3 Relatório de Produtividade – Projeto XP

Este documento apresenta as saídas do Sistema de Apoio ao Projeto, descrito na sessão 4.10. Tais saídas são:

- Horas estimadas por funcionalidade implementada;
- Horas gastas por funcionalidade implementada;
- Rendimento por iteração;
- Rendimento por programador;
- Atuação dos programadores.



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 153 CEP 05508-900 São Paulo SP
Telefone: (11) 3091-8583 Fax (11) 3091-5294

Departamento de Engenharia de Computação e Sistemas Digitais

SPM – Sistema Pré-anestésico Móvel

Especificação de Requisitos de Software

Autores: <i>Daniel Bonfim da Silva</i> <i>Rubens Jun Hirakawa</i>	Data de emissão: 13/08/2005
Revisor: <i>Sérgio Enrique Garbati Kreiner</i>	Data de revisão 14/8/2005



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 156 CEP 05508-900 São Paulo SP
Telefone: (11) 3091-5583 Fax (11) 3091-5294

Departamento de Engenharia de Computação e Sistemas Digitais

FOLHA DE CONTROLE DE REVISÕES

Número da versão	Data de emissão	Registro de modificações
00	8/8/2005	Versão inicial
01	11/8/2005	Inserção das descrições dos <i>story cards</i>
02	13/8/2005	Inserção dos casos de teste
03		
04		
05		
06		
07		
08		
09		
10		



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (11) 3091-5583 Fax (11) 3091-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Índice

1	Objetivo do Documento	4
2	Objetivo do Sistema	4
2.1	Nome do Sistema	4
2.2	Escopo	4
2.3	Definições, Siglas e Abreviaturas	4
3	Descrição Geral	5
3.1	Perspectivas do Produto	6
3.1.1	Interface com o Sistema	6
3.1.2	Interfaces de Usuário	6
3.1.3	Interfaces de Hardware	7
3.1.4	Interfaces com Software	7
3.1.5	Interfaces de Comunicação	7
3.2	Funções do Software	7
3.3	Características dos Usuários	7
3.4	Restrições	7
4	Requisitos Específicos	8
4.1	Story Cards	8
4.1.1	Log in / Log off	8
4.1.2	Inclusão de médico	8
4.1.3	Atualização de médico	8
4.1.4	Exclusão de médico	8
4.1.5	Inclusão de paciente	9
4.1.6	Atualização de paciente	9
4.1.7	Exclusão de paciente	9
4.1.8	Inclusão de ficha	9
4.1.9	Atualização de ficha	9
4.1.10	Exclusão de ficha	9
4.1.11	Carregamento de Pocket PCs	9
4.1.12	Descarregamento de Pocket PCs	9
4.2	Telas do sistema	10
4.3	Casos de teste	10
4.3.1	Log in / Log off	10
4.3.2	Inclusão de médico	10
4.3.3	Atualização de médico	11
4.3.4	Exclusão de médico	12
4.3.5	Inclusão de paciente	12
4.3.6	Atualização de paciente	12
4.3.7	Exclusão de paciente	13
		2



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (11) 3091-5583 Fax (11) 3091-5294

Departamento de Engenharia de Computação e Sistemas Digitais

4.3.8	Inclusão de ficha.....	14
4.3.9	Atualização de ficha.....	15
4.3.10	Exclusão de ficha.....	16
4.3.11	Carregamento de Pocket PCs.....	16
4.3.12	Descarregamento de Pocket PCs.....	17
4.4	Requisitos de Bancos de Dados.....	17



SPM – Sistema Pré-anestésico Móvel

Especificação de Requisitos de Software

1 Objetivo do Documento

O objetivo deste documento de Especificação de Requisitos do Software é identificar todas as funções que devem ser executadas pelo SPM bem como a estrutura necessária para a implementação deste.

Este é um documento técnico e, portanto, conta com nomenclaturas especiais que, quando necessário, serão explicadas na seção 2.3. *Definições, Siglas e Abreviaturas*.

2 Objetivo do Sistema

O sistema tem como objetivo o acesso que médicos fazem aos dados de pacientes que serão submetidos a anestésias através de *Pocket PCs*.

2.1 Nome do Sistema

SPM – Sistema Pré-anestésico Móvel

2.2 Escopo

O SPM servirá para agilizar as consultas que médicos anestesistas fazem aos dados de seus pacientes antes de ministrar os anestésicos; visa a substituir formulários em papel, usados correntemente, associando a portabilidade destes com a segurança e a organização de um computador móvel, entendendo-se por estes palms e pocket pcs, que serão os instrumentos usados para tais consultas. O sistema conta ainda com uma interface web, através da qual médicos e enfermeiras também podem cadastrar e consultar dados clínicos dos pacientes.

2.3 Definições, Siglas e Abreviaturas



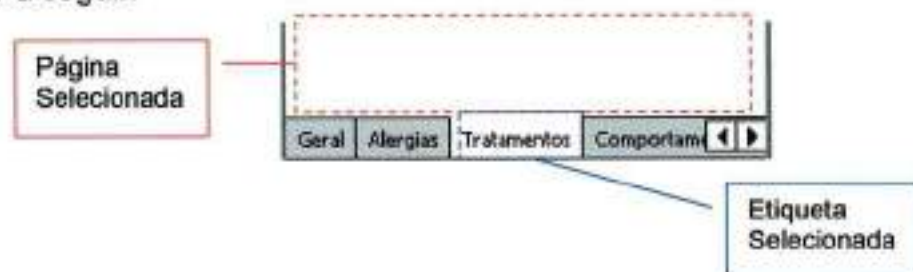
LAN – Local Area Network ou Rede Local

Pocket PC – computador de mão que usa o sistema operacional Windows CE.

SP – Sistema Pré-anestésico

SPD – Sistema Pré-anestésico para Desktops

Tab Control – Controle exibido em alguns dos formulários do SPM que divide os campos de preenchimento destes em páginas identificadas por etiquetas. Para acesso a cada uma das páginas, o usuário deve tocar em sua respectiva etiqueta. Um tab control é ilustrado a seguir:



3 Descrição Geral

O SPM é uma ferramenta para que médicos e enfermeiras possam cadastrar, consultar e alterar dados clínicos de pacientes através de dispositivos móveis (*handhelds*). Por tratar de dados sigilosos, o sistema contará com níveis de acesso diferentes para médicos, enfermeiras e administradores e todas as alterações nos dados clínicos dos pacientes serão armazenadas num histórico, sempre acompanhadas de seu autor. Além disso haverá criptografia de dados.

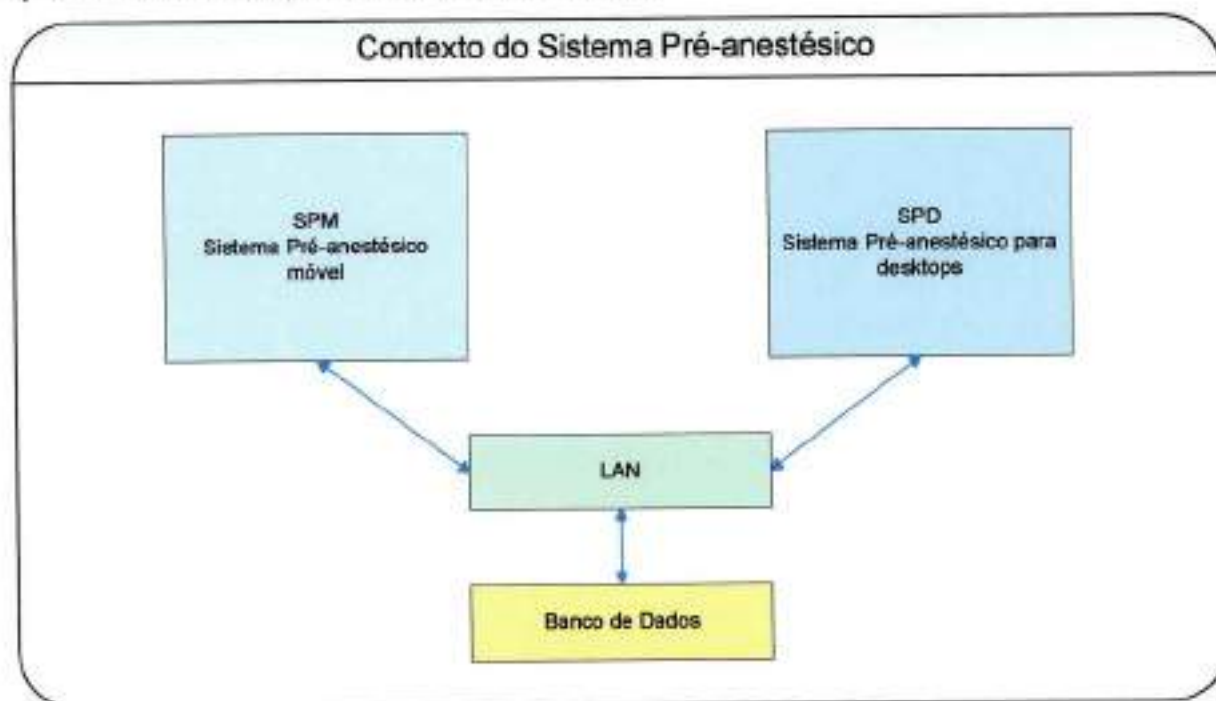
O sistema pré-anestésico móvel funcionará paralelamente a um sistema análogo para computadores convencionais de mesa (*desktops*) em rede, e ambos compartilharão um banco de dados.

As informações inseridas e alteradas nos dispositivos móveis do sistema pré-anestésico serão transferidas ao banco de dados quando estes forem conectados fisicamente à LAN do hospital.



3.1 Perspectivas do Produto

O sistema pré-anestésico móvel, conforme já foi mencionado, interage com um sistema análogo para computadores de mesa (desktops); o contexto de que ele faz parte é apresentado no diagrama de blocos a seguir:



O sistema pré-anestésico móvel compartilha um banco de dados com o sistema pré-anestésico para *desktops* mas ambos não trocam dados entre si.

3.1.1 Interface com o Sistema

O SPM não interage diretamente com o SPD, mas ambos são conectados à mesma LAN para acessarem o Banco de Dados do SP.

3.1.2 Interfaces de Usuário

O SPM terá uma interface adaptada para as telas pequenas de Pocket PCs.



3.1.3 Interfaces de Hardware

Os *Pocket PCs* integrantes do sistema necessitarão de infra estrutura (cabos ou dispositivos sem fio) para conexão à rede local. Os *desktops* devem estar conectados à rede local.

3.1.4 Interfaces com Software

O SPM é executado sobre o sistema operacional *Microsoft Windows Mobile 2003* com *.Net Framework* nos *Pocket PCs* e sobre o *Microsoft Windows XP* com *.Net Framework* nos *desktops*. O banco de dados usado será o *SQL Server 2000*.

3.1.5 Interfaces de Comunicação

No SPM a comunicação de *Pocket PCs* e *desktops* com o banco de dados é feita com a utilização de um protocolo de rede local com criptografia.

3.2 Funções do Software

O SPM permite o cadastro de pacientes, cadastro de fichas de anamnese para cada paciente e efetuar buscas por paciente ou ficha através do nome daquele ou código desta, ou por meio da data de cadastro da ficha.

3.3 Características dos Usuários

Os usuários do sistema são principalmente médicos, que exigem praticidade e eficiência do sistema. Eles geralmente não tem muito conhecimento computacional e usam apenas as funcionalidades básicas de seus dispositivos móveis. Também usam o sistema enfermeiros, com pouco conhecimento computacional.

3.4 Restrições

Os dados contidos no SPM são sigilosos e em função da ética médica devem ser protegidos de consultas alheias com senhas e criptografia. Os módulos do SPM para *Pocket PCs* e *desktops* devem ser desenvolvidos em *C#.Net*.
O Banco de Dados utilizado deve ser o *SQL Server 2000*.



Sob hipótese alguma os dados de anamnese serão apresentados incorretamente pelo sistema, pois este tipo de erro poderia resultar na prescrição de anestésicos incompatíveis com a condição física dos pacientes e isto poderia ser letal. Por isso o SPM deve contar com segurança nos acessos e nos tratamentos de erro (*security and safety*).

4 Requisitos Específicos

Esta sessão apresenta uma descrição detalhada dos requisitos do sistema, representados por Story Cards e Casos de Testes.

4.1 Story Cards

São apresentadas a seguir as descrições dos *story cards* do sistema:

4.1.1 Log in / Log off

Médicos devem ser capazes de acessar o SPM com seus Pocket PCs digitando seu nome de usuário e senha. Deve haver segurança no acesso: nenhum usuário com dados errados pode acessar o sistema.

4.1.2 Inclusão de médico

Administradores do sistema devem poder cadastrar médicos que poderão acessar o SPM com nome de usuário e senha próprios imediatamente no Pocket PC em que foi feito o cadastro em todo o sistema, assim que o Pocket for sincronizado

4.1.3 Atualização de médico

Administradores do sistema podem alterar os dados cadastrais dos médicos (nome, CRM, nome de usuário, senha).

4.1.4 Exclusão de médico

Administradores do SPM podem excluir médicos cadastrados, impedindo que estes tenham acesso ao sistema imediatamente no Pocket PC em que foi feita a exclusão e em todo o sistema, após a sincronização.



4.1.5 Inclusão de paciente

Médicos com acesso ao SPM poderão cadastrar novos pacientes.

4.1.6 Atualização de paciente

Médicos com acesso ao SPM poderão alterar os dados cadastrais dos pacientes.

4.1.7 Exclusão de paciente

Médicos com acesso ao SPM poderão excluir pacientes cadastrados no sistema.

4.1.8 Inclusão de ficha

Médicos com acesso ao SPM podem relacionar fichas aos pacientes cadastrados; as fichas serão inseridas em duas etapas: uma parte da consulta pré-anestésica e outra do histórico do paciente e planejamento da anestesia.

4.1.9 Atualização de ficha

Médicos podem atualizar fichas cadastradas via SPM. As fichas antigas vão para o histórico.

4.1.10 Exclusão de ficha

Médicos podem excluir fichas via SPM; estas vão para o histórico.

4.1.11 Carregamento de Pocket PCs

O médico com acesso ao SPW pode escolher os pacientes com que quer carregar o Pocket PC (SPM). Estes pacientes ficam bloqueados no SPW e para todos os demais médicos (podem ser acessados via SPW mas não carregados em outros SPM nem editados, de modo algum).

4.1.12 Descarregamento de Pocket PCs

Após usar os dados dos pacientes escolhidos, um médico deve descarregar seu Pocket PC para que os dados de tais pacientes sejam liberados no sistema (SPW). Alterações



em relação ao estado anterior devem ser salvas, após transferência das informações antigas para o histórico.

4.2 Telas do sistema

- (1) Tela de login.
- (2) Tela do menu principal;
- (3) Tela do menu de cadastro médico;
- (4) Tela do menu para cadastro de médicos;
- (5) Tela de busca por médico;
- (6) Tela do menu de cadastro de pacientes;
- (7) Tela do formulário para cadastro de pacientes (página "Dados" do tab control.
- (8) Tela de busca por paciente;
- (7) Tela do formulário para cadastro de pacientes (página "Fichas" do tab control);
- (10) Tela do formulário de consulta pré-anestésica;
- (11) Tela do histórico clínico e planejamento da anestesia.

4.3 Casos de teste

São apresentados a seguir os testes funcionais cuja correta execução é necessária para validar cada uma das histórias do cliente. As telas envolvidas em cada teste são referenciadas pela numeração apresentada na seção anterior.

4.3.1 Log in / Log off

Ao abrir o SPM o usuário verá a tela de login (1), onde deverá digitar seu nome de usuário e senha e clicar em "Entrar". Se o usuário for um médico cadastrado na base local do dispositivo móvel, será apresentada a tela do menu principal (2). Se o nome de usuário e a senha não estiverem cadastrados na base de dados local do dispositivo, será exibida a tela de login (1) com uma mensagem de falha na autenticação (A). Na tela do menu de cadastro médico (3) o usuário poderá tocar, a qualquer momento, em "Menu Principal" para retornar à tela do menu principal (2).

4.3.2 Inclusão de médico

Após o sucesso na autenticação, o usuário verá na tela do menu principal (2) a opção "Cadastro Médico"; ao tocar nela será mostrada a tela do menu de cadastro médico (3),



onde haverá a opção "Cadastrar Médico"; quando tocar nela o usuário verá a tela do formulário para cadastro de médicos (4). Após o preenchimento de todos os dados cadastrais o usuário pode tocar em "Confirmar" para inserir os dados do novo médico na base local; após a inserção será exibida a tela do menu de cadastro médico (3) com uma mensagem de sucesso na operação (B); se houver falha na inserção do novo médico, será exibida mensagem de falha na operação (M). Se houver falha no preenchimento de algum dos campos, será exibida a tela do formulário para cadastro de médicos (4) com uma mensagem indicando os erros encontrados (C). A qualquer momento o usuário poderá tocar o botão "Cancelar" na tela do formulário para cadastro de médicos (4) para retornar à tela do menu de cadastro médico (3) sem relizar nenhuma mudança na base de dados local. Na tela do menu de cadastro médico (3) o usuário poderá tocar, a qualquer momento, em "Menu Principal" para retornar à tela do menu principal (2).

4.3.3 Atualização de médico

Após o sucesso na autenticação, o usuário verá na tela do menu principal (2) a opção "Cadastro Médico"; ao tocar nela será mostrada a tela do menu de cadastro médico (3), onde haverá a opção "Atualizar Médico". Ao tocar nela será exibida a tela de busca por médico (5) com o botão "Excluir" invisível; nela haverá uma caixa combo com todos os nomes dos médicos cadastrados em ordem alfabética, onde o usuário poderá escolher o nome do médico cujos dados deseja alterar, e uma caixa de texto para a inserção do CRM do médico desejado. Após escolher o nome ou inserir o CRM, o usuário deverá clicar em "Buscar" para que seja exibida a tela do formulário para cadastro de médicos (4) com os dados do médico escolhido no passo anterior. A qualquer momento, o usuário poderá tocar em "Cancelar" na tela de busca por médico (5) para retornar à tela do menu de cadastro médico (3). Na tela do formulário para cadastro de médicos (4), após alterar os dados necessários, o usuário deverá tocar em "Confirmar" para que a base de dados local seja alterada com os novos dados do médico selecionado; após a alteração da base de dados será exibida a tela do menu de cadastro médico (3) com uma mensagem de sucesso na operação (B); se houver falha na atualização dos dados do médico selecionado, será exibida mensagem de falha na operação (M). Se houver falha no preenchimento de algum dos campos, será exibida a tela do formulário para cadastro de médicos (4) com uma mensagem indicando os erros encontrados (C). A qualquer momento o usuário poderá tocar em "Cancelar" na tela do formulário para cadastro de médicos (4) para voltar à tela do menu de cadastro médico (3) sem que nenhuma mudança na base de dados principal seja executada. Na tela do menu de cadastro médico (3) o usuário poderá tocar, a qualquer momento, em "Menu Principal" para retornar à tela do menu principal (2).



4.3.4 Exclusão de médico

Após o sucesso na autenticação, o usuário verá na tela do menu principal (2) a opção "Cadastro Médico"; ao tocar nela será mostrada a tela do menu de cadastro médico (3), onde haverá a opção "Excluir Médico". Ao tocar nela será exibida a tela de busca por médico (5) com o botão "Excluir" habilitado e visível; nela haverá uma caixa combo com todos os médicos cadastrados em ordem alfabética, onde o usuário poderá escolher o nome do médico cujos dados deseja alterar, e um campo para a inserção do CRM do médico desejado. Após escolher o nome ou inserir o CRM, o usuário deverá clicar em "Excluir". Será exibida uma caixa de mensagem para que o usuário confirme ou não a exclusão (D); se o usuário confirmar, aquele registro será excluído da base de dados local; caso contrário, nenhuma operação será realizada na base de dados local e será exibida a tela de busca por médico (5) com o botão "Excluir" habilitado e visível. A qualquer momento, o usuário poderá tocar em "Cancelar" na tela de busca por médico (5) para retornar à tela do menu de cadastro médico (3). Na tela do menu de cadastro médico (3) o usuário poderá tocar, a qualquer momento, em "Menu Principal" para retornar à tela do menu principal (2).

4.3.5 Inclusão de paciente

Após o sucesso na autenticação, o usuário verá na tela do menu principal (2) a opção "Cadastro de Pacientes"; ao tocar nela será mostrada a tela do menu de cadastro de pacientes (6). Na tela do menu de cadastro de pacientes (6) o usuário poderá tocar, a qualquer momento, em "Menu Principal" para retornar à tela do menu principal (2). Na tela do menu de cadastro de pacientes (6) haverá a opção "Cadastrar Paciente"; quando tocar nela o usuário verá a tela do formulário para cadastro de pacientes (7) com a página "Dados" do *tab control* selecionada. Após o preenchimento de todos os dados cadastrais o usuário pode tocar em "Confirmar" para inserir os dados do novo paciente na base local; após a inserção será exibida a tela do menu de cadastro de pacientes (6) com uma mensagem de sucesso na operação (E); se houver falha na inserção dos dados do novo paciente será exibida uma mensagem de falha na operação (N). Se houver falha no preenchimento de algum dos campos, será exibida a tela do formulário para cadastro de pacientes (7) com uma mensagem indicando os erros encontrados (F). A qualquer momento o usuário poderá tocar o botão "Cancelar" na tela do formulário para cadastro de pacientes (7) para retornar à tela de busca por paciente (8) sem relizar nenhuma mudança na base de dados local. A qualquer momento, o usuário poderá tocar em "Menu Principal" na tela de busca por paciente (8) para retornar à tela do menu principal (2).



4.3.6 Atualização de paciente

Após o sucesso na autenticação, o usuário verá na tela do menu principal (2) a opção "Cadastro de Pacientes"; ao tocar nela será mostrada a tela do menu de cadastro de pacientes (6). Na tela do menu de cadastro de pacientes (6) o usuário poderá tocar, a qualquer momento, em "Menu Principal" para retornar à tela do menu principal (2). Na tela do menu de cadastro de pacientes (6) onde haverá a opção "Atualizar Paciente". Ao tocar nela será exibida a tela de busca por paciente (8) com o botão "Excluir" invisível e o botão "Buscar" visível e habilitado; nela haverá uma caixa combo com todos os nomes dos pacientes cadastrados em ordem alfabética, onde o usuário poderá escolher o nome do paciente cujos dados deseja alterar, e uma caixa de texto para a inserção do CPF do paciente cujos dados se deseja alterar. Após escolher o nome ou inserir o CPF, o usuário deverá clicar em "Buscar" para que seja exibida a tela do formulário para cadastro de pacientes (7) com os dados do paciente escolhido no passo anterior e com a página "Dados" do *tab control* selecionada. A qualquer momento, o usuário poderá tocar em "Menu Principal" na tela de busca por paciente (8) para retornar à tela do menu principal (2). Na tela do formulário para cadastro de pacientes (7), após alterar os dados necessários, o usuário deverá tocar em "Confirmar" para que a base de dados local seja alterada com os novos dados do paciente selecionado; após a alteração da base de dados será exibida a tela do menu de cadastro de pacientes (6) com uma mensagem de sucesso na operação (E); se houver falha na atualização dos dados do paciente selecionado será exibida uma mensagem de falha na operação (N). Se houver falha no preenchimento de algum dos campos, será exibida a tela do formulário para cadastro de pacientes (7) com uma mensagem indicando os erros encontrados (F). A qualquer momento o usuário poderá tocar em "Cancelar" na tela do formulário para cadastro de pacientes (7) para voltar à tela de busca por paciente (8) sem que nenhuma mudança na base de dados principal seja executada.

4.3.7 Exclusão de paciente

Após o sucesso na autenticação, o usuário verá na tela do menu principal (2) a opção "Cadastro de Pacientes"; ao tocar nela será mostrada a tela do menu de cadastro de pacientes (6). Na tela do menu de cadastro de pacientes (6) o usuário poderá tocar, a qualquer momento, em "Menu Principal" para retornar à tela do menu principal (2). Na tela do menu de cadastro de pacientes (6) haverá a opção "Excluir Paciente". Ao tocar nela será exibida a tela de busca por paciente (8) com o botão "Excluir" habilitado e visível e o botão "Buscar" visível e habilitado; nela haverá uma caixa combo com todos os pacientes cadastrados em ordem alfabética, onde o usuário poderá escolher o nome do paciente cujos dados deseja excluir, e um campo para a inserção do CPF do paciente desejado. Após escolher o nome ou inserir o CPF, o usuário deverá clicar em "Excluir". Será exibida uma caixa de mensagem para que o usuário confirme ou não a exclusão



(G); se o usuário confirmar, aquele registro será excluído da base de dados local; caso contrário, nenhuma operação será realizada na base de dados local e será exibida a tela de busca por paciente (8) com o botão "Excluir" habilitado e visível. A qualquer momento, o usuário poderá tocar em "Menu Principal" na tela de busca por paciente (8) para retornar à tela do menu principal (2).

4.3.8 Inclusão de ficha

Após o sucesso na autenticação, o usuário verá na tela do menu principal (2) a opção "Fichas"; ao tocar nela será exibida a tela de busca por paciente (8) com o botão "Excluir" invisível e o botão "Buscar" visível e habilitado. A qualquer momento, o usuário poderá tocar em "Menu Principal" na tela de busca por paciente (8) para retornar à tela do menu principal (2). Na tela de busca por paciente (8) haverá uma caixa combo com todos os nomes dos pacientes cadastrados em ordem alfabética, onde o usuário poderá escolher o nome do paciente ao qual deseja associar uma nova ficha, e uma caixa de texto para a inserção do CPF do paciente ao qual deseja associar uma nova ficha. Após escolher o nome ou inserir o CPF, o usuário deverá clicar em "Buscar" para que seja exibida a tela do formulário para cadastro de pacientes (7) com a página "Fichas" do *tab control* selecionada; nesta página o usuário deve tocar em "Nova Ficha" para que seja exibida a tela do formulário de consulta pré-anestésica (10) com os dados do paciente selecionado anteriormente. Na tela de consulta pré-anestésica (10) o usuário poderá preencher todos os campos que desejar; os campos de preenchimento estarão distribuídos em páginas acessíveis por um *tab control*. Na tela do formulário de consulta pré-anestésica (10), a qualquer momento, o usuário poderá tocar em "Voltar" para retornar à tela do formulário para cadastro de pacientes (10), com a página "Fichas" do *tab control* selecionada, sem realizar nenhuma mudança na base de dados local. Na tela do formulário de consulta pré-anestésica (10), a qualquer momento o usuário poderá tocar em "Salvar" para armazenar a nova ficha, com todos os campos por ele preenchidos, na base de dados local; após a atualização da base de dados local será exibida a tela do histórico clínico e planejamento da anestesia (11) com uma mensagem de sucesso na operação (I); se houver falha na inserção dos dados da consulta pré-anestésica da nova ficha, será exibida uma mensagem de falha na operação (Q). Os dados serão salvos como uma nova ficha. O usuário poderá preencher todos os campos que desejar; os campos de preenchimento estarão distribuídos em páginas acessíveis por um *tab control*. Na tela do histórico clínico e planejamento da anestesia (11), a qualquer momento, o usuário poderá tocar em "Voltar" para retornar à tela do formulário de consulta pré-anestésica (10) sem realizar nenhuma mudança na base de dados local. Na tela do histórico clínico e planejamento da anestesia (11), a qualquer momento, o usuário poderá tocar em "Salvar" para armazenar a nova ficha, com todos os campos por



ele preenchidos, na base de dados local e partir para a tela do formulário para cadastro de pacientes (7) com a página "Fichas" do *tab control* selecionada e com uma mensagem de sucesso na operação (L); se houver falha na inserção dos dados do histórico clínico e planejamento da anestesia da nova ficha, será exibida uma mensagem de falha na operação (O). Na tela do formulário para cadastro de pacientes (7) o usuário poderá tocar, a qualquer momento, em "Cancelar" para retornar à tela de busca por paciente (8).

4.3.9 Atualização de ficha

Após o sucesso na autenticação, o usuário verá na tela do menu principal (2) a opção "Fichas"; ao tocar nela será exibida a tela de busca por paciente (8) com o botão "Excluir" invisível e o botão "Buscar" visível e habilitado. A qualquer momento, o usuário poderá tocar em "Menu Principal" na tela de busca por paciente (8) para retornar à tela do menu principal (2). Na tela de busca por paciente (8) haverá uma caixa combo com todos os nomes dos pacientes cadastrados em ordem alfabética, onde o usuário poderá escolher o nome do paciente cuja ficha deseja atualizar, e uma caixa de texto para a inserção do CPF do paciente cuja ficha deseja atualizar. Após escolher o nome ou inserir o CPF, o usuário deverá clicar em "Buscar" para que seja exibida a tela do formulário para cadastro de pacientes (7) com a página "Fichas" do *tab control* selecionada; nesta página haverá uma tabela cujas colunas serão "Nº da Ficha", "Médico", "Atualizar" (coluna de botões) e "Excluir" (coluna de botões); para atualizar uma determinada ficha, o usuário deverá tocar em "Atualizar" na linha desta ficha; será exibida então a tela do formulário de consulta pré-anestésica (10) com os dados da ficha selecionada. Na tela do formulário de consulta pré-anestésica (10), a qualquer momento, o usuário poderá tocar em "Voltar" para retornar à tela do menu de fichas (9). Na tela do formulário de consulta pré-anestésica (10), a qualquer momento, o usuário poderá tocar em "Salvar" para armazenar na base de dados local os dados atualizados da ficha escolhida e partir para a tela do histórico clínico e planejamento da anestesia (11) com uma mensagem de sucesso na operação (I); se houver falha na inserção dos dados da consulta pré-anestésica da nova ficha, será exibida uma mensagem de falha na operação (Q). Os dados serão salvos como uma nova ficha. Os dados originais serão mandados para a tabela de histórico da base de dados local e deixarão de ser exibidos pelo SPM. O usuário poderá preencher todos os campos que desejar; os campos de preenchimento estarão distribuídos em páginas acessíveis por um *tab control*. Na tela do histórico clínico e planejamento da anestesia, a qualquer momento, o usuário poderá tocar em "Voltar" para retornar à tela do formulário de consulta pré-anestésica (10) sem realizar nenhuma mudança na base de dados local. A qualquer momento o usuário poderá tocar em "Salvar" para armazenar a nova ficha, com todos os campos por ele preenchidos, na base de dados local e partir para a tela do formulário para cadastro de pacientes (7) com os dados do paciente cuja



ficha foi atualizada, a página "Fichas" do *tab control* habilitada e com uma mensagem de sucesso na operação (L); se houver falha na atualização dos dados da ficha selecionada, será exibida uma mensagem de falha na operação (O). Na tela do formulário para cadastro de pacientes (7) o usuário poderá tocar, a qualquer momento, em "Cancelar" para retornar à tela de busca por paciente (8).

4.3.10 Exclusão de ficha

Após o sucesso na autenticação, o usuário verá na tela do menu principal (2) a opção "Fichas"; ao tocar nela será exibida a tela de busca por paciente (8) com o botão "Excluir" invisível e o botão "Buscar" visível e habilitado. A qualquer momento, o usuário poderá tocar em "Menu Principal" na tela de busca por paciente (8) para retornar à tela do menu principal (2). Na tela de busca por paciente (8) haverá uma caixa combo com todos os nomes dos pacientes cadastrados em ordem alfabética, onde o usuário poderá escolher o nome do paciente cuja ficha deseja excluir, e uma caixa de texto para a inserção do CPF do paciente cuja ficha deseja excluir. Após escolher o nome ou inserir o CPF, o usuário deverá clicar em "Buscar" para que seja exibida a tela do formulário para cadastro de pacientes (7) com a página "Fichas" do *tab control* selecionada; nesta página haverá uma tabela cujas colunas serão "Nº da Ficha", "Médico", "Atualizar" (coluna de botões) e "Excluir" (coluna de botões); para excluir uma determinada ficha, o usuário deverá tocar em "Excluir" na linha desta ficha; uma mensagem para confirmação da exclusão (J) será exibida; se a exclusão for confirmada os dados da ficha serão encaminhados para uma tabela de histórico na base de dados local e deixarão de ser exibidos pelo SPM; após a atualização da base de dados local será exibida a tela do formulário para cadastro de pacientes (7) com os dados do paciente cuja ficha foi excluída, a página "Fichas" habilitada e com uma mensagem de sucesso na operação (L); se houver falha na exclusão da ficha selecionada será exibida uma mensagem de falha na operação (O). Se a exclusão não for confirmada, nenhuma operação será exibida na base de dados local e será exibida a tela do formulário para cadastro de pacientes (7) com os dados do paciente selecionado inicialmente e a página "Fichas" do *tab control* habilitada. Na tela do formulário para cadastro de pacientes (7) o usuário poderá tocar, a qualquer momento, em "Cancelar" para retornar à tela de busca por paciente (8).

4.3.11 Carregamento de Pocket PCs

Após o sucesso na autenticação, o usuário verá na tela do menu principal (2) a opção "Carregar"; antes de tocar nela o usuário deve conectar seu *Pocket PC* ou seu *Palm* à rede do servidor de banco de dados principal. Após clicar em "Sincronização" aguardar



um período de processamento, o usuário receberá uma mensagem de sucesso na sincronização (R).

4.3.12 Descarregamento de Pocket PCs

Após o sucesso na autenticação, o usuário verá na tela do menu principal (2) a opção "Descarregar"; antes de tocar nela o usuário deve conectar seu *Pocket PC* ou seu *Palm* à rede do servidor de banco de dados principal. Após clicar em "Sincronização" a aguardar um período de processamento, o usuário receberá uma mensagem de sucesso na sincronização (R).

4.4 Requisitos de Bancos de Dados

Os dados armazenados no sistema são informações cadastrais dos pacientes, informações de acesso de médicos e fichas de anamnese dos pacientes. Os dados de acesso de médicos e enfermeiras são usados permanentemente enquanto que os dados cadastrais e fichas de pacientes são acessados geralmente no período pré-cirúrgico. Espera-se que os dados de acesso sejam solicitados inúmeras vezes ao dia por cada usuário, enquanto o uso de fichas e dados cadastrais de cada paciente deve se repetir muitas vezes ao dia no período pré-cirúrgico mas será nulo em outras ocasiões. Entretanto, por questões legais e para controle dos médicos, que desejam manter um histórico dos seus pacientes, todos os dados cadastrais e fichas destes serão mantidos no banco de dados permanentemente. Fichas alteradas serão salvas como novos registros, para que se constitua o histórico e para evitar perdas.



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 n° 158 CEP 05508-900 São Paulo SP
Telefone: (11) 3091-5583 Fax (11) 3091-5294

Departamento de Engenharia de Computação e Sistemas Digitais

SPW – Sistema Pré-anestésico *Web*

Especificação de Requisitos de Software

Autores: <i>Daniel Bonfim da Silva</i> <i>Rubens Jun Hirakawa</i>	Data de emissão: <i>25/9/2005</i>
Revisor: <i>Sérgio Enrique Garbati Kreiner</i>	Data de revisão <i>26/9/2005</i>



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 153 CEP 05508-900 São Paulo SP
Telefone: (11) 3091-5583 Fax (11) 3091-5294

Departamento de Engenharia de Computação e Sistemas Digitais

FOLHA DE CONTROLE DE REVISÕES

Número da versão	Data de emissão	Registro de modificações
00	25/9/2005	Versão inicial
01	26/9/2005	Correção de erros verificados na revisão
02		
03		
04		
05		
06		
07		
08		
09		
10		



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (11) 3091-5583 Fax (11) 3091-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Índice

1	Objetivo do Documento	3
2	Requisitos Específicos	3
2.1	Story Cards	3
2.1.1	Log in / Log off	3
2.1.2	Inclusão de médico	4
2.1.3	Atualização de médico	4
2.1.4	Exclusão de médico	4
2.1.5	Inclusão de paciente	4
2.1.6	Atualização de paciente	4
2.1.7	Exclusão de paciente	4
2.1.8	Inclusão de ficha	4
2.1.9	Atualização de ficha	4
2.1.10	Exclusão de ficha	5
2.1.11	Alteração de senha	5
2.2	Telas do sistema	5
2.3	Casos de teste	5
2.3.1	Alteração de senha	5



SPW – Sistema Pré-anestésico Web

Especificação de Requisitos de Software

1 Objetivo do Documento

Este documento é complementar ao documento de especificação de requisitos de software do Sistema Pré-anestésico Móvel (SPM), já que o Sistema Pré-anestésico Web (SPW) é um outro módulo do sistema a que pertence o SPM.

Na especificação de requisitos de software do SPM, refere-se ao SPW como Sistema Pré-anestésico para Desktops (SPD). Após uma revisão conceitual deste sistema, decidiu-se chamá-lo de Sistema Pré-anestésico Web (SPW), pois ele pode ser executado em *desktops* e em *laptops*, sempre conectados à rede (web) hospitalar.

São apresentados neste documento os requisitos específicos do SPW.

2 Requisitos Específicos

Esta sessão apresenta uma descrição detalhada dos requisitos do sistema, representados por Story Cards e Casos de Testes.

2.1 Story Cards

São apresentadas a seguir as descrições dos *story cards* do sistema:

As histórias do SPW são análogas às do SPM, com exceção da alteração de senha, que não é disponibilizada no primeiro.

2.1.1 Log in / Log off

Médicos, enfermeiras e administradores devem ser capazes de acessar o SPW. Deve haver segurança.



2.1.2 Inclusão de médico

Administradores do SPW devem ser capazes de cadastrar novos médicos.

2.1.3 Atualização de médico

Administradores do SPW devem ser capazes de atualizar dados dos médicos cadastrados no sistema.

2.1.4 Exclusão de médico

Administradores do SPW devem ser capazes de excluir dados de médicos cadastrados no sistema.

2.1.5 Inclusão de paciente

Médicos e enfermeiros cadastrados no SPW devem ser capazes de incluir dados de novos pacientes no sistema

2.1.6 Atualização de paciente

Médicos e enfermeiros cadastrados no SPW devem ser capazes de atualizar dados dos pacientes cadastrados no sistema

2.1.7 Exclusão de paciente

Médicos e enfermeiros podem excluir dados dos pacientes cadastrados no sistema (que são transferidos para a base de histórico)

2.1.8 Inclusão de ficha

Médicos e enfermeiros devem ser capazes de incluir novas fichas, sempre associadas a pacientes já cadastrados, em duas etapas: uma da consulta pré-anestésica e outra do histórico do paciente e planejamento da anestesia.



2.1.9 Atualização de ficha

Médicos e enfermeiros devem ser capazes de atualizar qualquer campo de cada parte da ficha médica.

2.1.10 Exclusão de ficha

Médicos são capazes de excluir fichas cadastradas do sistema. Estas serão transferidas para a base de histórico.

2.1.11 Alteração de senha

Enfermeiros e médicos devem ser capazes de alterar a própria senha de acesso.

2.2 Telas do sistema

As telas do SPW são análogas às do SPM.

No entanto, a tela seguinte é exclusiva ao SPW.

- (12) Tela do formulário para alteração de senha.

O SPW não tem botões de sincronização no menu principal.

2.3 Casos de teste

Nesta seção é apresentado o teste funcional do SPW que não é análogo aos do SPM. Para validação das histórias descritas nas seções 2.1.1 a 2.1.10 deste documento é necessário executar os testes descritos entre as seções 4.3.1 e 4.3.10 do documento de especificação de requisitos do SPM.

2.3.1 Alteração de senha

Após o sucesso na autenticação, o usuário verá na tela do menu principal (2) a opção "Alterar Senha"; ao tocar nela será mostrada a tela do formulário para alteração de senha (12). Nesta tela, o usuário deverá digitar sua senha atual, confirmá-la, digitar a nova senha e tocar em "Confirmar" para realizar a mudança de senha no banco de dados e retornar à tela do menu principal (2). A qualquer momento, na tela do formulário para



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (11) 3091-5583 Fax (11) 3091-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Relatório de Produtividade

Projeto XP

Autores:

Daniel Bonfim da Silva

Rubens Jun Hirakawa

Sérgio Enrique Garbati Kreiner



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (11) 3091-5583 Fax (11) 3091-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Índice

1	Objetivo do Documento	2
2	Rendimento por Iteração	2
2.1	Primeira Iteração	2
2.2	Segunda Iteração	2
2.3	Terceira Iteração	3
2.4	Gráfico de rendimento	3
2.5	Conclusão	3
3	Atuação dos programadores	4
3.1	Atuação: programador principal	4
3.2	Atuação: revisor	4
3.3	Conclusão	4



1 Objetivo do Documento

Este documento apresenta as informações de produtividade do desenvolvimento de software realizado como parte do Projeto XP.

Os números aqui apresentados foram extraídos do Sistema de Apoio ao Projeto XP.

2 Rendimento por Iteração

Neste capítulo são apresentadas as tabelas de horas estimadas, horas gastas e rendimento em cada iteração de desenvolvimento no Projeto XP.

2.1 Primeira Iteração

Primeira Iteração			
Story Card	Estimativa (h)	Gastas (h)	Rendimento
login no SPM	9	11	81,82%
Atualização de Ficha	10	10	100,00%
Carregamento de Pocket PCs	10	8	125,00%
Descarregamento de Pocket PCs	10	8	125,00%
Totais	39	37	
Rendimento da Iteração	105,41%		

2.2 Segunda Iteração

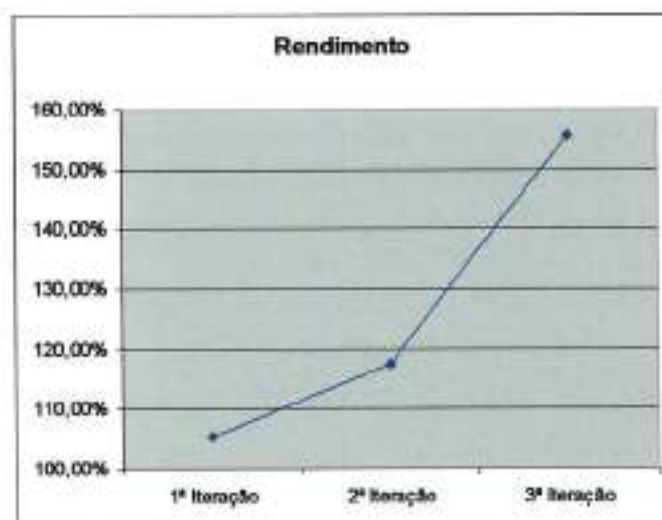
Segunda Iteração			
Story Card	Estimativa (h)	Gastas (h)	Rendimento
Log in/Log off no SPW	10	8	125,00%
Inclusão de Médico no SPW	4	3	133,33%
Atualização de Médico no SPW	4	3	133,33%
Exclusão de médico no SPW	4	3	133,33%
Inclusão de Paciente no SPW	4	3	133,33%
Atualização de Paciente no SPW	3	3	100,00%
Exclusão de Paciente no SPW	3	3	100,00%
Inclusão de ficha no SPW	6	7	85,71%
Atualização de Ficha no SPW	5	4	125,00%
Exclusão de ficha no SPW	4	3	133,33%
Totais	47	40	
Rendimento da Iteração	117,50%		



2.3 Terceira Iteração

Terceira Iteração			
Story Card	Estimativa (h)	Gastas (h)	Rendimento
login no SPM	6	4	150,00%
Alteração de senha	5	4	125,00%
Carregamento de Pocket PCs	5	3	166,67%
Atualização de Ficha	7	4	175,00%
Descarregamento de Pocket PCs	5	3	166,67%
Totais	28	18	
Rendimento da Iteração	155,56%		

2.4 Gráfico de rendimento



2.5 Conclusão

O rendimento aumentou significativamente entre as iterações; cresceu 12,09% entre a primeira e a segunda iteração e 38,06% entre a segunda e a terceira. Este é um reflexo da absorção das mudanças culturais determinadas pelo XP.



O número médio de horas gastas por iteração foi de 32, um número pertinente para uma semana de trabalho (com 8 horas diárias).

3 Atuação dos programadores

Neste capítulo são apresentados o número total de horas em que cada programador atuou como programador principal ou revisor, além do rendimento atingido por cada um deles em cada atuação.

3.1 Atuação: programador principal

Atuação: Programador Principal			
	Estimativa (h)	Gastas (h)	Rendimento
Consultor	18	20	90,00%
Prog 1	56	44	127,27%
Prog 2	40	31	129,03%

3.2 Atuação: revisor

Atuação: Revisor			
	Estimativa (h)	Gastas (h)	Rendimento
Prog 1	54	47	114,89%
Prog 2	60	48	125,00%

3.3 Conclusão

Nota-se que o rendimento dos programadores envolvidos no projeto foi maior que o do consultor, porque estes assimilaram as mudanças culturais determinadas pelo XP.

O Programador 1 e o Programador 2 renderam mais quando atuaram como programadores principais. O programador principal ganha por compartilhar a resolução de dúvidas técnicas com o revisor.