

Projetos Participativos no Contexto Brasileiro

Leonardo P. Tizzei¹, Fernanda Foschiani², Alberto Santos²

¹Instituto de Computação– Universidade Estadual de Campinas
Campinas, SP, Brasil.

²Departamento de Engenharia de Petróleo – Universidade Estadual de Campinas
Campinas, SP, Brasil

tizzei@ic.unicamp.br, {fernanda.foschiani,alberto.santos}@dep.fem.unicamp.br

***Abstract.** The trend towards studying participatory projects in computer science comes from the growing presence of technology in our daily lives. This has led to a growing interest in applications that offer facilities to users and promote users engagement during its development. Nevertheless, these features are particularly challenging in Brazilian context where there is a lack of knowledge and material related to technology. This work aims in presenting Brazilian participatory projects that engages user in different phases during the software development cycle.*

***Resumo.** A tendência de estudar projetos participativos na computação advém da crescente presença de tecnologia no cotidiano. Isso tem provocado interesse em aplicações que ofereçam facilidade de utilização e engajamento dos usuários durante seu desenvolvimento. Contudo, essas características são particularmente desafiadoras para o contexto brasileiro, onde existe uma carência de conhecimento e material em relação à tecnologia. Esse trabalho visa apresentar projetos participativos realizados no Brasil, que envolvem o usuário em diferentes fases do ciclo de desenvolvimento de software.*

1. Introdução

A tendência de a tecnologia tornar-se cada vez mais onipresente tem provocado um aumento no interesse de empresas e governos em projetos de sistemas de software cujo objetivo seja a facilidade de acesso e utilização pelo usuário e um maior engajamento deste [9]. Desta forma, cresce o interesse em disciplinas, técnicas e metodologias que visam alcançar esses objetivos através da participação do usuário durante o desenvolvimento do sistema.

Dentro do contexto do desenvolvimento de software no Brasil a onipresença da tecnologia torna-se ainda mais desafiadora para os desenvolvedores de sistemas, uma vez que grande parte da população tem um restrito conhecimento e familiaridade com tecnologias mais recentes. Segundo dados do PNAD 2005 [10], 21% da população com 10 anos ou mais acessaram a internet nos 3 meses que antecederam a entrevista. Na faixa de rendimento de até $\frac{1}{4}$ do salário mínimo *per capita*, o percentual de pessoas que acessaram a internet era de 3,3%. Outra informação importante, é que apenas 18,6% dos domicílios possuíam um computador e, dentro desses, 13,7% dos computadores tinham acesso à Internet. Essas informações refletem o restrito acesso que boa parcela da

população tem em relação a essas tecnologias.

Portanto, torna-se importante a aproximação entre o usuário e a tecnologia para que o produto final seja aceito e usado com sucesso. O envolvimento do usuário pode ocorrer em diferentes fases do ciclo de desenvolvimento do produto. Algumas técnicas de design participativo auxiliam o usuário a participar nas fases iniciais do desenvolvimento. Já metodologias como a Bazar [11] permitem que usuários com habilidades técnicas contribuam na fase de desenvolvimento do sistema.

O objetivo deste artigo é apresentar trabalhos recentes e relevantes na área de projeto participativo dentro do contexto de desenvolvimento de software no Brasil. São apresentadas metodologias e técnicas que foram aplicadas em diferentes fases do ciclo de desenvolvimento de software.

O restante deste artigo será apresentado da seguinte forma: a Seção 2 apresenta técnicas e metodologias utilizadas em projetos participativos. A Seção 3 mostra uma visão geral das fases de um projeto participativo. Em seguida, na Seção 4, detalhamos alguns projetos participativos que foram aplicados no Brasil. Por fim, na Seção 5 apresentamos as considerações finais sobre este trabalho.

2. Fundamentos de Projeto Participativo

Pensando no ambiente de projeto participativo, é importante avaliar algumas características, tais como: características de planejamento, design (projeto), metodologia de desenvolvimentos, entre outras.

2.1 Planejamento Participativo

Um dos aspectos mais importantes na gestão de projetos é o planejamento. Apesar de um bom planejamento não ser suficiente para o sucesso do projeto, ele contribui de forma significativa para evitar seu fracasso.

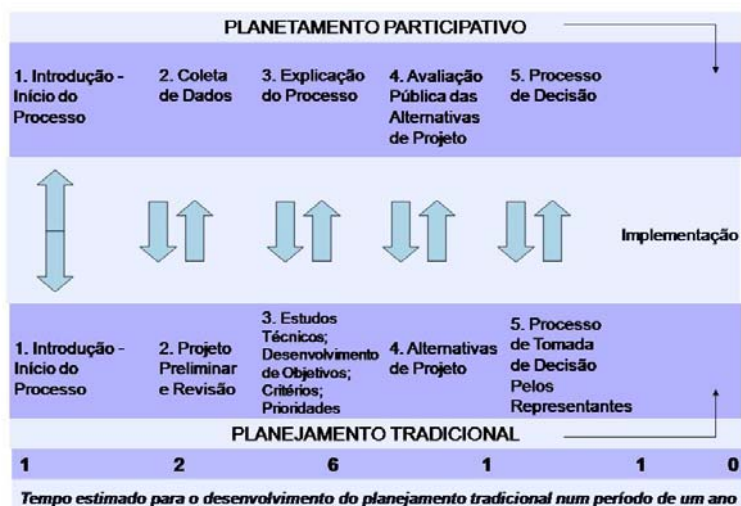
De acordo com Rozental [13], em projetos de Tecnologia de Informação e Comunicação (TIC) é sabido que os usuários aceitarão o novo sistema mais facilmente quando envolvidos em seu projeto desde o princípio. É também sabido que este tipo de abordagem aumenta o tempo de realização do projeto, tornando as fases do projeto mais lentas, principalmente as fases de desenho e a codificação.

Connor [15] argumenta que o conceito de participação construtiva em projetos pode ser formulado como: “um processo sistemático que fornece uma oportunidade de cidadãos, planejadores, gerentes e representantes eleitos para compartilhar a sua experiência, conhecimento e objetivos e combinar a sua energia de criar um plano que é tecnicamente sólido, economicamente atraente, geralmente entendido e aceito pela maior parte dos afetados por ele e assim politicamente viável.”.

Uma das principais diferenças entre o planejamento tradicional e planejamento participativo é que no segundo, cada stakeholder contribui no processo de discussão pois pode existir uma grande variedade de opiniões, idéias, conhecimentos, entre outros. As metas poderão ser avaliadas mais globalmente e sob diferentes pontos de vista. Isto irá aumentar o processo de planejamento como um todo mas serão tratados aspectos essenciais à continuidade e aceitação do projeto. Podem-se citar alguns pontos relevantes do planejamento participado: stakeholders com diferentes interesses; modificação no processo de planejamento; maior interação entre os stakeholders.

A Figura 1 ilustra tanto o planejamento participativo quanto o planejamento tradicional na visão de [16].

Figura 1. Processo de Planejamento Participativo



2.2 Técnicas de Design Participativo

Personas é o nome dado à técnica de design participativo para aumentar o engajamento dos usuários. As “personas” são pessoas fictícias, que possuem nome, ocupação, família, amigos etc. Também possuem idade, gênero, nível educacional, status sócio-econômico, histórias e metas vida.

Os defensores desta técnica argumentam que pessoas fictícias exercem atração sobre as reais e citam como exemplo a popularidade que novelas, filmes e séries têm em vários países. Outro fator importante é que quanto maior o tempo de exposição às personas, mais “reais” e complexas elas se tornam.

A técnica personas permite que mais aspectos de comportamento sejam analisados do que a utilização de cenários, por exemplo. Isso porque é inerente ao indivíduo criar expectativas e fazer previsões sobre as pessoas que nos cercam, assim como as personas também.

Contudo, é necessário que a criação das personas seja baseada em um bom estudo qualitativo e quantitativo sobre os usuários. Esse estudo deve ficar disponível para todos os participantes do projeto, para que as personas possam ganhar credibilidade entre eles.

Cenários. A abordagem de desenvolvimento com a utilização de **cenários** encoraja o envolvimento dos usuários e provê um vocabulário comum entre as pessoas que estão participando do desenvolvimento do projeto. Além disso, cenários ajudam a antecipar problemas na execução de uma determinada tarefa [6].

Um cenário é uma descrição que contém (i) atores, (ii) informações sobre o histórico do ator, (iii) as metas dos atores e (iv) uma seqüência de ações e eventos. Algumas aplicações permitem que uma dessas informações seja omitida.

Cenários podem ser expressos em várias formas e tipos de mídia. Por exemplo, cenários podem ser narrativas textuais, *storyboards* e video mockups. Além disso,

cenários podem ser escritos em uma linguagem formal, semi-formal ou informal. A Figura abaixo (extraída de Rosson [3]) mostra um exemplo de cenário informal, um tipo de cenário usado frequentemente para visualizar tarefas dos usuários.

Etnografia é uma técnica de observação que pode ser usada para entender requisitos organizacionais e sociais [5]. Um analista é imerso no ambiente onde o sistema será usado e observa o cotidiano de trabalho e faz anotações sobre como são realizadas determinadas tarefas. A etnografia ajuda a descobrir requisitos implícitos do sistema ao invés dos requisitos formais de um processo.

Protótipos são versões iniciais do sistema que são usadas para demonstrar conceitos, tentar opções de interface gráfica e descobrir problemas e suas possíveis soluções [5]. O desenvolvimento rápido de protótipos é essencial para controlar gastos e permitir que o usuário tenha um contato com o software nas fases iniciais de desenvolvimento. Protótipos servem também para descobrir e validar requisitos do sistema.

2.3 Licenças de Software

Software Livre, segundo o conceito criado pela Free Software Foundation, é qualquer programa de computador que pode ser usado, copiado, estudado, modificado e redistribuído com algumas restrições, e deve atender os quatro tipos de liberdade, citadas mais adiante, para seus usuários [5].

A Free Software Foundation (FSF) é uma organização sem fins lucrativos, fundada em 1985 por Richard Stallman, e dedica-se à eliminação de restrições sobre a cópia, redistribuição, entendimento e modificação de programas de computadores.

As quatro liberdades citadas são [5]:

- A liberdade para executar o programa, para qualquer propósito (liberdade nº. 0);
- A liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades (liberdade nº. 1). Acesso ao código-fonte é um pré-requisito para esta liberdade;
- A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo (liberdade nº. 2);
- A liberdade de aperfeiçoar o programa, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie (liberdade nº. 3). Acesso ao código-fonte é um pré-requisito para esta liberdade.

Assim sendo, para que um programa seja considerado um Software Livre, seus usuários, sejam pessoas físicas ou jurídicas, devem receber todas estas liberdades. Ou seja, podem distribuir cópias, com ou sem modificações, cobrando-se ou não uma taxa pela distribuição, sem ser necessária a autorização ou pagamento para fazê-la. Além disso, deve-se ter a liberdade de modificações e uso do código, não havendo a necessidade de citação de referência. Um Software Livre pode ser usado em qualquer sistema operacional, com qualquer finalidade.

Por causa do problema com a palavra “free” na língua inglesa, as pessoas tinham

concepções erradas em relação ao Software Livre. A confusão do termo “free” na língua inglesa resulta de um fator semântico e um fator econômico, explicados a seguir.

O fator semântico se deve ao fato de que a palavra “free” pode significar tanto liberdade quanto gratuidade. Richard Stallman alerta que “free” deve ser entendido como em “free speech” (liberdade de expressão), e não como em “free beer” (cerveja grátis).

Já o fator econômico diz respeito à gratuidade dos Softwares Livres, já citado anteriormente. A maior parte dos Softwares Livres pode ser obtida gratuitamente através da Internet. Isto é um efeito indireto da liberdade de redistribuição. Como o Software Livre pode ser redistribuído por qualquer pessoa, ele perde seu valor de venda. Portanto, Software Livre não é “não-comercial”. Mas, sua comercialização através do modelo tradicional de venda de licenças de uso é difícil.

As liberdades concedidas pelo Software Livre independem de sua gratuidade. Assim sendo, o fato de se cobrar ou não pela distribuição ou pela licença de uso do software não implica diretamente em ser Software Livre ou não. Um Software Livre pode ser copiado e vendido, mesmo que não sejam feitas modificações. Ou seja, Software Livre não necessariamente precisa ser gratuito. Por outro lado, existem programas distribuídos gratuitamente que não podem ser modificados nem redistribuídos e assim, não são Softwares Livres.

Código Aberto (tradução de Open Source) é o termo usado pela Open Source Initiative (OSI), criada em 1998, para se referir aos Softwares Livres.

Para ser considerado um Código Aberto, o software precisa permitir o acesso ao seu código fonte (e não necessariamente acompanhado das "4 liberdades" do Software Livre), e deve atender aos 10 critérios da Definição de Código Aberto, que incluem itens como Livre Redistribuição, Permissão de Trabalhos Derivados, Não Discriminação e Distribuição da Licença [<http://fatalnet.com/index.php/download-linux/>], entre outros.

2.4 Software Livre X Código Aberto

As definições "oficiais" de Software Livre e de Código Aberto são ligeiramente diferentes. A definição de Software Livre é considerada mais rigorosa, mas as licenças de Código Aberto que não são consideradas licenças de software livre são geralmente obscuras.

A maioria das licenças que atende à já mencionada Definição de Software Livre (FSF) também atende à Definição de Código Aberto (OSI). Assim sendo, pode-se dizer que, na maior parte das vezes, se um determinado software é livre, ele também é de código aberto, e vice-versa. A diferença prática entre as duas entidades está em seus objetivos, filosofia e modo de agir, e não nos softwares ou licenças [5].

A maior diferença entre os dois movimentos é o fato de a Free Software Foundation condenar o Software Proprietário [5]. O movimento do Software Livre é social. Ele julga que não é ético aprisionar conhecimento científico, que deve estar sempre disponível, para permitir assim a evolução da humanidade. Já o Movimento pelo Código Aberto, que não é um movimento social, mas é voltado ao mercado, prega que o software desse tipo traz diversas vantagens técnicas e econômicas.

2.5 Metodologias Catedral e Bazar

Catedral é a metodologia de desenvolvimento típica de software proprietário. Quando usada esta metodologia, o código fonte é avaliado em cada atualização do software, mas o código desenvolvido entre duas atualizações é restrito ao grupo de desenvolvedores [11].

O método Catedral é amplamente difundido e é baseado na concepção tradicional de desenvolvimento de software. Suas principais características são:

- Desenvolvimento de software de forma controlada e centralizada;
- Amplamente utilizada tanto na indústria quanto na vida acadêmica;
- Necessita de um arquiteto central, que leva toda a responsabilidade do processo de desenvolvimento.

Bazar é a metodologia caracterizada pelo desenvolvimento do software de maneira totalmente aberta e disseminada. Este estilo parte do princípio de que qualquer pessoa pode contribuir para o desenvolvimento de um programa e as melhores contribuições serão mantidas no código. Uma de suas principais características é que problemas são facilmente encontrados, pois como citado por Eric Raymond, "com muitos olhos, todos os 'bugs' (erros) serão visíveis" [11].

As principais características da metodologia Bazar são:

- Ter um objetivo claro e de fácil motivação;
- Ter um bom líder que tem as funções de manter a comunidade motivada, não deixando que o objetivo se perca e tomar as medidas necessárias para que tudo funcione;
- Ter uma comunidade de participantes, trabalhando de forma totalmente descentralizada;
- Ter um ambiente de trabalho dinâmico, que forneça meios de comunicação e disseminação do código (como a Internet e ferramentas como sites e listas de discussões);

Bazar x Catedral

Na metodologia Catedral, os intervalos de liberação do software são longos pois quando encontrados erros e problemas de desenvolvimento, leva-se muito tempo para serem examinados, corrigidos e testados. Esta demora se deve ao fato de o grupo de desenvolvimento ser geralmente pequeno e as versões liberadas são restritas a este grupo.

Já na metodologia Bazar, assume-se que erros são geralmente um fenômeno trivial. Os problemas são expostos para toda a comunidade de usuários e desenvolvedores e por isso é corrigido rapidamente. Assim, as liberações de novas versões acontecem com bastante frequência.

A Tabela 1. *Principais diferenças entre o Modelo Catedral e o Modelo Bazar* expõe as principais diferenças entre as duas metodologias de desenvolvimento.

Tabela 1. Principais diferenças entre o Modelo Catedral e o Modelo Bazar

Modelo Catedral	Modelo Bazar
Trabalho controlado por um grupo de projetistas	Não há hierarquia entre os participantes, apenas um líder
Estabelecimento de metodologias, tarefas e prazos	Projetos informalmente organizados ao redor de uma proposta inicial
Demora para lançamento de atualizações	Atualizações constantes
Dificuldade para atingir <i>massa crítica</i>	Participação voluntária e conseqüentemente <i>massa crítica</i> atingida rapidamente
Dificuldade para atingir qualidade esperada	Qualidade atingida rapidamente

3. Visão Geral de Projetos Participativos

De acordo com o PMBOK [17], projetos são temporários (tem início, meio e fim) e únicos (cada projeto é diferente do outro).

Assim, um projeto pode ser definido em termos de suas características distintas, onde um projeto é um empreendimento temporário com o objetivo de criar um produto ou serviço único. Temporário significa que cada projeto tem um começo e um fim bem definidos. Único significa que o produto ou serviço produzido é de alguma forma diferente de todos os outros produtos ou serviços semelhantes.

3.1 Principais componentes

De acordo com o PMBOK [17], podemos dizer que os principais "componentes" para a gestão de projetos são:

- Escopo: incluem os processos necessários para garantir que o projeto inclua todas as atividades requeridas e somente estas.
- Tempo: incluem os processos necessários para garantir que o projeto seja realizado no prazo correto.
- Custo: incluem os processos necessários para garantir que o projeto será concluído com o capital aprovado.
- Qualidade: incluem os processos necessários para garantir que o projeto irá satisfazer as necessidades para os quais foi projetado.
- Recursos: incluem os processos necessários para fazer uso efetivo das pessoas envolvidas no projeto.
- Comunicação: incluem os processos necessários para garantir que o projeto tenha a devida geração, disseminação, armazenamento e disponibilidade das informações.
- Risco: incluem os processos necessários para sistematicamente

identificar, analisar e mitigar os riscos do projeto.

- **Aquisição:** incluem os processos necessários para adquirir mercadorias e serviços.
- **Integração:** incluem os processos necessários para garantir que os diversos componentes do projeto serão coordenados de acordo.

Estes componentes formam a base de todas as áreas de conhecimento do gerenciamento de projeto.

3.2 Ambiente de projeto

De acordo com Rozendal [13] um projeto participativo não deve ser visto como um fenômeno isolado, pois “forças externas” têm um fator de influência consideravelmente grande sobre o projeto. Na abordagem de “sistema fechado”, os resultados do projeto são produtos relativamente simples e com alto grau de qualidade técnica, além disto, o aceite para adoção do sistema é obtido após o encerramento do projeto. Já na abordagem de “sistema aberto”, a influência externa é grande e a adoção do projeto é fortemente relacionada ao grau de aceitação do ambiente. Neste caso entende-se que o processo de desenvolvimento deveria ser visto como um processo de interação com o ambiente externo altamente complexo. Portanto, os resultados têm um balanço entre qualidade e aceitação pelo ambiente. Isto implica numa redução do controle sobre o projeto.

Usando o modelo de Cusworth (1993), as forças externas podem ser divididas em 2 tipos: "fatores culturais" e "atores políticos". Ambas são difíceis de direcionar e invés de tratá-las diretamente, a arte do gerenciamento neste caso reside no fato de prevê-las antecipadamente.

Influencia política

Atores políticos são todas as organizações que tem o potencial de influenciar o curso do projeto. Embora o gerenciamento do projeto tenha algum poder de direcioná-las, não é possível controlar todo o ambiente. Neste caso, a arte do gerenciamento de projeto reside na capacidade de criar um projeto que englobe os diferentes interesses dos stakeholders.

Influencia cultural

Fatores culturais são todos os padrões do comportamento coletivo que influenciam o projeto, mas que são impossíveis de controlar através do gerenciamento do projeto. Neste caso, a arte do gerenciamento reside na habilidade da organização de projeto ser inventiva o suficiente para se moldar a forma de trabalho neste ambiente.

3.3 Ciclo de Vida

Existem vários tipos e variantes de ciclos de vida de desenvolvimento de software, dentre ele podemos citar: cascata; incremental, espiral, prototipação, evolucionário, embutido, etc. Cada tipo com suas características e abordagens peculiares. Segundo Rozental [13], uma modelo para TIC seria dividir o processo de desenvolvimento em 7 fases denominadas: “*Visualization*”, “*Formulation*”, “*Appraising*”, “*Mobilisation*”, “*Implementation*”, “*Operation*”, “*Transfer and*

evaluation”. Resumidamente falando, estas fases podem ser descritas como:

Visualization: A primeira fase inicia com uma imagem da situação desejada. Esta fase resulta numa descrição dos objetivos do projeto, da especificação e idéia geral da estratégia para a realização do projeto.

Formulation: Nesta fase o foco é em como realizar o projeto, incluindo requisitos funcionais e não funcionais, o tempo e recursos necessários. Esta fase tem como saída um plano que permite uma tomada de decisão sobre o início das demais fases.

Appraising: Após a especificação, é possível aos stakeholders e analistas financeiros a tomada de decisão sobre o direcionamento do projeto no sentido de continuar ou encerrá-lo. O resultado desta fase é um documento formal, tal como contrato.

Mobilisation: Esta fase inicia com a preparação de um plano operacional detalhado. Isto guiará o gerenciamento com os passos detalhados que deverão ser realizados imediatamente e em menor detalhe os passos que deverão ser realizados posteriormente. O resultado aqui é um projeto com todos os membros da equipe ativamente envolvidos.

Implementation: A construção do projeto inicia nesta fase. O modelo proposto não menciona qual metodologia a ser utilizada, mas isto pode ser feito usando uma das duas metodologias citadas anteriormente:

- Abordagem Catedral – a implementação do desenho final (abordagem catedral) quando se sabe exatamente as especificações do projeto e como realizá-las.
- Abordagem Bazar – quando os objetivos do projeto são claramente conhecidos, mas as especificações exatas e como realizá-las não são, além dos recursos envolvidos, a abordagem bazar pode ser usada. A vantagem deste método é o envolvimento de pessoas da comunidade, a redução de recursos e manutenção, além de poder ajudar aos usuários especificar melhor o que quer.

Operation: Neste momento o sistema pode ser colocado em operação. Nesta fase os usuários deverão ser treinados, os manuais dos usuários deverão ser finalizados e erros (bugs) no código deverão ser resolvidos.

Transfer and evaluation: Após o projeto ter sido colocado em operação com sucesso, a responsabilidade sobre o produto final é transferida para os usuários.

4. Projetos participativos no contexto brasileiro

Nesta seção serão mostrados alguns projetos participativos recentes no Brasil.

4.1 Caleidoscópio Júnior

O uso da internet por crianças têm crescido continuamente [1]. Surge então a necessidade de desenvolver sites específicos para esse público, de forma a atender as especificidades das crianças. Alguns pesquisadores destacam que as crianças devem ter acesso à informação, proteção e participação na mídia [2]. Essa participação de crianças

na construção do conteúdo em outras mídias, como rádio e televisão, têm sido incentivado [3].

Como forma de dar às crianças a oportunidade para o desenvolvimento de sites para elas mesmas, Martins et al. [1] utilizou de técnicas de design participativo para criar um portal infantil chamado **Caleidoscópio Júnior** (<http://www.nied.unicamp.br/~caleidojr/>). Essa abordagem visa dar voz ao público infantil na criação de tecnologia, ao invés de deixar essa tarefa para um adulto.

Para envolver as crianças no desenvolvimento do portal, foram realizadas atividades de design participativo com crianças entre 6 e 10 anos, meninos e meninas com diferentes níveis de acesso a tecnologia. Durante as atividades de design participativo as crianças criaram protótipos do portal em papel, além de navegarem em sites infantis e utilizarem uma ferramenta de bate-papo desenvolvida para crianças.

Durante as atividades, foi observado que muitas crianças já possuíam certa familiaridade com a utilização do computador e da internet. Segundo Martins et al. [1], admitir a importância da tecnologia no universo infantil provoca mudanças no processo educacional, no sentido de incorporar os recursos tecnológicos de comunicação e interação nas atividades que são propostas em tais processos. Contudo, no trabalho de Martins et al., nenhuma observação é feita a respeito de crianças (ou pessoas) que nunca tiveram acesso a computadores ou a internet.

Algumas crianças conseguiram fazer relações entre os objetos reais e sua representação virtual, o que demonstra uma familiaridade com a tecnologia. No caso, o objeto em questão era o caleidoscópio. Elas também sabiam interagir com o objeto no mundo virtual de forma que a obter um efeito parecido com o objeto real. Por exemplo, num dos relatos uma criança sugere que para girar o caleidoscópio virtual, bastaria clicar num botão.

Outra prova da familiaridade das crianças com a tecnologia e a internet mais especificamente foram os desenhos muito parecidos com os menus laterais vistos em muitos sites.

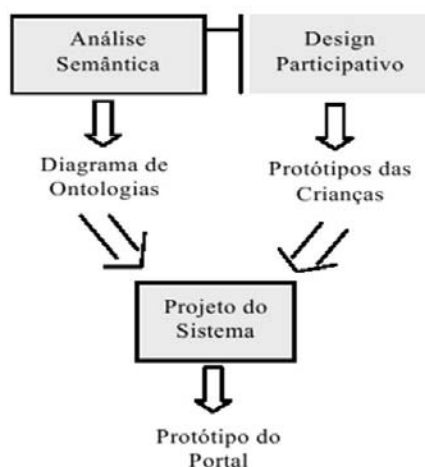
Contudo, o desenvolvimento do site não é necessariamente feito somente a partir de protótipos de interface. Melo et al. [4] utiliza ontologias¹ para representar os elementos que fazem parte do site.

Segundo as autoras, um diagrama de ontologia representa o entendimento e a expressão do designer para o espaço infantil na internet. Por exemplo, entre esses elementos estão a criança, a internet, a língua entre outros. A Figura 2 mostra como os resultados da atividade resultaram no protótipo do portal infantil.

O projeto do portal infantil Caleidoscópio Júnior mostrou que não só é possível a participação de crianças no desenvolvimento tecnologias através de técnicas de design participativo. Ou seja, mesmo pessoas cuja familiaridade com tecnologia não é grande e cujo desenvolvimento intelectual não é completo puderam participar satisfatoriamente no desenvolvimento de um portal para a internet. Portanto, é possível prover condições para que usuários possam participar de forma mais ativa no desenvolvimento de tecnologias a serem usadas por eles.

¹ Ontologia é um modelo de dados que representa um conjunto de conceitos e o relacionamentos entre eles dentro de um domínio [5].

Figura 2. Etapas para a criação do protótipo



4.2 e-poupatempo

O **Poupatempo** é um serviço público criado pelo governo do estado de São Paulo que visa atender o cidadão de forma eficiente e com qualidade. O **e-poupatempo** é outra iniciativa para prover alguns dos serviços do Poupatempo eletronicamente. Os serviços são oferecidos em salas de atendimento equipadas com computadores e assistidas por monitores.

Foi realizada uma parceria entre o governo paulista e o Laboratório de Interação Homem-Computador (LabiHC) da Universidade de São Paulo (USP). Como resultado desta parceria foi criado um conjunto de diretrizes para o desenvolvimento de serviços públicos para o governo do estado de São Paulo [6]. Além disso, foi analisado como é feita a modelagem dos usuários do e-poupatempo.

O envolvimento de usuários no desenvolvimento de ferramentas pressupõe o conhecimento do perfil destes usuários e sua representatividade dentro do conjunto de usuários [7]. No desenvolvimento de um sistema de software para governo eletrônico a preocupação com a usabilidade torna-se ainda maior, não só pela enorme variedade de usuários, mas também pelo fato que esse sistema é muitas vezes a única opção que o usuário tem para realizar determinada tarefa. Por exemplo, para declarar o imposto de renda no Brasil existe apenas um software que deve ser utilizado por todos, gostem do software ou não.

Foram coletados dados sobre os usuários que utilizavam as salas de atendimento do e-poupatempo através dos monitores. Além de descrever dados sobre si mesmo, o usuário relatava dificuldades na usabilidade dos serviços. Foram respondidos cerca de 8000 questionários com o objetivo de coletar informações sobre o usuário, como idade, nível de escolaridade, faixa salarial entre outros.

A técnica selecionada para modelar o perfil dos usuários foi personas. Como visto no momento 2, **persona** é uma tipificação do usuário baseada nos comportamentos e metas destes. Diferentes métodos como estatística e entrevistas podem ser utilizados na criação das personas.

A extração de informação da base de dados sobre os usuários foi realizada

através de uma ferramenta chamada Weka [8]. A ferramenta **Weka** é usada para identificação dos agrupamentos de dados através de técnicas de classificação, *clustering* e regras de associação. Com auxílio da Weka, foi possível identificar o perfil dos usuários sobre diferentes características. Por exemplo, a **Tabela 2** mostra a distribuição da escolaridade entre os usuários de e-poupatempo. Outras tabelas similares foram construídas, em relação à faixa salarial, idade entre outras características.

Tabela 2 - Amostra com o nível de escolaridade dos usuários de e-poupatempo

<i>Nível de escolaridade</i>	<i>Qtde</i>
alfabetizado	100
1º grau incompleto	702
1º grau completo	571
2º grau incompleto	1998
2º grau completo	3818
Superior incompleto	953
Superior completo	741
Pós-graduação	146
Não respondeu	33
Respostas inválidas	108

Através dos dados obtidos com esses agrupamentos, foram criadas 10 personas que juntas representam o conjunto de usuários de e-poupatempo. Cada persona possui um grau de representatividade, que é importante na determinação das *features* dos sistemas. Por exemplo, se uma persona possui representatividade de 23% ela tem maior influência que outra que tem 15%. A partir das características obtidas pelo agrupamento, também foram criadas descrições do comportamento de cada persona, bem como sua expectativa e satisfação frente ao uso do governo eletrônico.

O trabalho de Filgueiras et al. [7] afirma que a utilização de personas é um meio de prover usabilidade para sistemas de governo eletrônico como o e-poupatempo. Contudo, não foram apresentados avaliações sobre os possíveis aperfeiçoamentos obtidos através da técnica de personas.

4.3 Incluir-SIM

Para um projeto de Economia Solidária é necessária a utilização de um software que seja capaz de mapear a produção e o consumo local, rápida e eficientemente. Atualmente não existe nenhum Software Livre com tais características que pudesse ser adaptável para diferentes comunidades.

O objetivo do projeto Incluir-SIM é justamente o desenvolvimento de um Software Livre que realize a análise e a tabulação dos dados de um mapeamento de consumo e produção. E esta é a primeira etapa para a implantação de um empreendimento solidário, gerado numa comunidade [12]. O programa está sendo desenvolvido por um grupo de pesquisa da Universidade Federal da Bahia em parceria com a Associação de Fomento a Economia Solidária, a Bansol. Estes grupos adotaram a metodologia Bazar para o desenvolvimento do software.

Além da tabulação, análise e cruzamento de dados, este software servirá como um banco de dados de consulta para orientar as ações dentro das mais diversas comunidades.

O Incluir-SIM é um sistema web e tem como objetivo a realização de mapeamentos de produção e consumo nas variadas comunidades do Brasil. Com o software será possível realizar, de maneira eficiente, flexível e precisa, a tabulação e análise dos dados oriundos dos mapeamentos e o cruzamento destes dados. Assim ele possibilita que as comunidades construam uma análise detalhada do seu próprio perfil socioeconômico, criando, portanto, condições para o desenvolvimento sustentável de redes locais de economia solidária nos territórios envolvidos.

De acordo com as informações contidas no site do projeto, <http://twiki.im.ufba.br/bin/view/IncluirSim/EditalProjeto>, o software será disponibilizado nos repositórios oficiais de Software Livre e em toda a Rede Brasileira de Economia Solidária.

As principais características que marcam o desenvolvimento usando a metodologia Bazar são as citadas abaixo:

- A existência de um objetivo claro;
- Uma grande comunidade de participantes e desenvolvedores;
- Comunidade motivada;
- Não existe um grupo restrito de desenvolvedores e integrantes do grupo (qualquer pessoa pode colaborar com o projeto);
- Ambiente de trabalho dinâmico, o qual provê ferramentas para a comunicação entre os membros;
- Código disponibilizado nos repositórios oficiais de software livre.

O grupo desenvolvedor do software acredita que a disponibilização do código nos repositórios trará diversos benefícios, citando os principais: dinamização do processo de mapeamento de produção e consumo de comunidades, auxílio no direcionamento da criação de empreendimentos produtivos, diminuição dos custos de produção dos empreendimentos solidários, e também evitará a competição entre os integrantes da rede.

5. Considerações Finais

Este artigo procurou mostrar, em linhas gerais, os fundamentos, metodologias e alguns exemplos de projetos participativos no contexto brasileiro, com envolvimento maior do usuário em diferentes aspectos do projeto, desde o planejamento, projeto (design) até mesmo no desenvolvimento de software e validação dos resultados. Apesar de insipiente, pois estes tipos de projetos ainda são particularmente desafiadores no Brasil, eles estão aos poucos tomando vulto. Isto faz-nos crer que num futuro próximo haverá um engajamento maior tanto por parte dos financiadores quanto por parte da sociedade, de um modo geral, pois aos poucos o abismo ente a carência de conhecimento e material em relação à tecnologia vem sendo reduzido.

De acordo com o que foi mostrado neste artigo, observamos que nos projetos participativos o “ambiente externo” tem uma influência muito significativa exigindo

técnicas e metodologias que diferem dos projetos tidos como tradicionais.

Apesar de muito estudo sobre este assunto ainda existe muito que ser explorado e muito a ser analisado, onde este artigo procurou trazer algumas contribuições e achados para as questões de projetos participativos no contexto brasileiro.

6. Referências

- [1] Martins, Maria Cecília; Melo, Amanda M.; Baranauskas, M. Cecília C. (2003). "Participação de crianças na concepção de um Portal Infantil na Internet". In: Anais do XXIII Congresso da Sociedade Brasileira de Computação, Volume V, IX WIE, pp. 461 – 472, ISBN: 85-88442-56-6 (v. 5).
- [2] Carlsson, Ulla; Feilitzen, Cecilia Von. "A criança e a mídia: imagem, educação, participação" UNESCO, Ed. Cortez . 2002. ISBN: 85-249-0831-9
- [3] Radio pela infância. URL: [<http://www.unicef.org/brazil/radio.htm>]
- [4] Melo, A. M.; Baranauskas, M. C. (2003). "Construindo um Portal para Crianças com Crianças: Uma Abordagem Participativa ao Design". In: ICECE 2003 - International Conference on Engineering and Computer Education. ISBN: 85-89120-08-2 (CD-ROM).
- [5] Wikipedia. URL: [www.wikipedia.org]
- [6] Guia de Usabilidade dos Serviços Públicos e-poupatempo. URL: [<http://www.cqgp.sp.gov.br/manuais/usabilidade/usabilidade.htm#4>]
- [7] Filgueiras, L.; Júnior, P.; Sakai, R.; Filho, A.; Torres, C.; Barbarian, I. (2005) "Personas como modelo de usuários de serviços de governo eletrônico". In: Latin American conference on Human-computer interaction, vol.124, pp.319-324.
- [8] WEKA, Weka 3: Data mining software in Java, *The University of Waikato*, 2005. URL: [<http://www.cs.waikato.ac.nz/ml/weka/>]
- [9] Merkel C. et al. Participatory Design in Community Computing Contexts: Tales From the Field. PDC 04: Proceedings of the eighth conference on Participatory design 2004.
- [10] PNAD 2005. URL: [<http://www.ibge.gov.br/home/estatistica/populacao/trabalhoerendimento/pnad2005/default.shtm>]
- [11] Raymond, Eric Steven; "The Cathedral and the Bazaar"
- [12] Universidade Federal da Bahia – Projeto Onda Digital – Incluir-SIM. URL: [<http://twiki.im.ufba.br/bin/view/IncluirSim/WebHome>]
- [13] Rozendal, R. Cultural and Political Factors in the Design of ICT Projects in Developing Countries – Research Report, nº 13, March 2003.
- [14] Connor, D.M. Constructive Citizen Participation: A Resource Book, Victoria: Development Press, (Revised Edition 1994).
- [15] Connor, D.M. "From partisans to partners," *Community Planning Review*, 22(1), (1972).
- [16] Connor D. M. (s/d) Preventing and Resolving Public Controversy. URL: [<http://www.connor.bc.ca/connor/preventing.html>]
- [17] PMBOK. URL: [www.pmi.org]