

Uma Visão Geral Sobre a Ferramenta Plandora

Por Alberto Paterlini Pereto

Este documento contém a visão geral da ferramenta Plandora e das funcionalidades implementadas. Para cada funcionalidade são fornecidas as motivações para a sua criação bem como as limitações e como isso está sendo endereçado.

O projeto para criação da ferramenta **Plandora** foi iniciado em dezembro de 2003 com a seguinte motivação: Criar uma ferramenta que suprisse a necessidade de controle dos recursos de projetos (preferencialmente de software, mas não obrigatoriamente).

Para isso, foram definidos alguns pré-requisitos que estão listados abaixo em forma de funcionalidades. A maioria já está disponível na última versão da ferramenta.

Controle de Solicitações

1. ***FUNCIONALIDADE: Criação de um SPOC (Single Point of Contact). Um canal de entrada de solicitações dos usuários (internos ou externos) evitando a perda ou ruído na informação.***

MOTIVAÇÃO: sobretudo em empresas onde o cliente está próximo do recurso da tarefa, é comum clientes "cortarem caminho" e fazerem pedidos diretamente, causando ruído e descontrole. É muito difícil ao gerente de projeto estimar prazos e planejar alocação de recursos se parte do que esta sendo realizado chega sem o seu conhecimento. Além disso, a criação de um SPOC possibilita a obtenção de estatísticas (quantitativa e qualitativa) das solicitações dos projetos, bem como a tomada de ações corretivas ou estratégicas acerca delas.

2. ***FUNCIONALIDADE: O canal de entrada deveria permitir a visibilidade histórica das solicitações dos clientes***

MOTIVACAO: É importante que o cliente tenha acesso a todas as solicitações feitas e saiba de que forma a solicitação foi tratada, isto é, para cada etapa do tramite da solicitação quais pessoas foram envolvidas, quando e que comentários pertinentes foram adicionados.

3. *FUNCIONALIDADE: A ferramenta deveria ser multi-projetos, e os usuários cadastrados deveriam poder fazer parte de um ou mais projetos.*

MOTIVACAO: Nas empresas é muito comum um recurso trabalhar em projetos em paralelo, ou mesmo fazer overlap entre projetos, isto é, enquanto está testando um projeto, já iniciou o levantamento de requisitos de outro. Sendo assim, o gerenciamento de multi-projetos com intersecção de recursos era um requisito básico.

4. *FUNCIONALIDADE: A ferramenta deveria possibilitar a estruturação de projetos hierarquicamente.*

MOTIVAÇÃO: projetos e subprojetos são bastante comuns nas empresas. Às vezes um projeto cresce ou é expandido, e acaba-se criando subprojetos ou módulos complementares. A idéia aqui era poder visualizar os projetos de forma hierárquica. Na pratica, se um projeto tivesse uma determinada KPI ou Relatório, este elemento também faria parte automaticamente do projeto "pai", ou, se um usuário fosse líder do projeto pai, ele teria acesso aos dados de todos os projetos filhos.

5. *FUNCIONALIDADE: Cada usuário deveria ter um perfil atrelado a cada projeto, basicamente os perfis seriam: líder, recurso ou cliente. Porém um mesmo usuário deveria ser capaz, por exemplo, de ser líder em um projeto e cliente em outro. Cada perfil possibilitaria visualizar determinadas funcionalidades.*

MOTIVAÇÃO: Era importante criar perfis simples, porém bem definidos aos usuários da ferramenta. Isso facilitaria o rápido entendimento sobre o escopo de cada componente do projeto (que para pequenos projetos e pequenas empresas é fundamental) e também serviria como uma forma de educar os usuários na realização do trabalho com processo.

LIMITAÇÃO: ainda a ferramenta não possibilita a criação de perfis customizados. Embora a ferramenta tenha um cadastro de funções e possibilite, a partir da tela de projetos, definir para cada usuário do projeto uma ou mais destas "funções" (por exemplo: DBA, Project Manager, Scrum Master, etc), estes papeis ou funções ainda não afetam em nada o controle de acesso a determinadas funcionalidades.

6. *FUNCIONALIDADE: Ao requisitar uma solicitação, a ferramenta deveria ser capaz de prover meios para que o líder pudesse avaliar a viabilidade da requisição, rejeitá-la (se necessário) ou aceitá-la (alocando um ou mais recursos do projeto).*

MOTIVAÇÃO: Outra questão importante é em relação ao escopo da solicitação. As solicitações que chegam por escrito podem ser dimensionadas de forma mais eficiente do que as que chegam verbalmente. Em projetos com pouca margem para atrasos é fundamental que o líder tenha controle sobre as solicitações que estão entrando no sistema. Além disso, ele alivia stress nos recursos que poderão se concentrar integralmente na execução da tarefa, já que as solicitações dos clientes já foram previamente filtradas, negociadas e dimensionadas.

7. *FUNCIONALIDADE: A ferramenta deveria possibilitar a separação dos prazos dos clientes com os dos recursos. Ao aceitar uma solicitação e fazer a alocação, o líder poderia opcionalmente prometer um prazo para o cliente, mas que seria totalmente desvinculado do planejamento interno das tarefas dos recursos, dando a possibilidade ao líder de criar um buffer de segurança entre a data prometida ao cliente e a data final cobrada dos recursos.*

MOTIVAÇÃO: Atualmente um conceito bastante forte em gerência de projetos são as técnicas de Critical-Chain. Essa é uma forma de trabalho que dá ao líder ou gerente de projetos, meios para se trabalhar com milestones diferentes entre o cliente e os recursos. Assim, de um lado ele ganha uma margem de segurança para minimizar a visibilidade de atrasos vista pelo cliente e por outro lado ele pode passar datas mais realistas (sem gordura) ao recurso, pois essa técnica segue o princípio da "Lei do Estudante" que diz que os recursos de um projeto sempre começam lentamente e dão um *sprint* no final do prazo de entrega independente do tempo que foi estimado.

LIMITAÇÃO: É necessário implementar no Gantt Chart da ferramenta recursos mais sofisticados para edição e visualização dos *buffers* de segurança.

8. **FUNCIONALIDADE:** O sistema deveria permitir a customização do formulário de solicitações para agregar informações específicas de determinados projetos.

MOTIVAÇÃO: Cada projeto possui as suas especificidades e, sobretudo no formulário de solicitações que é a porta de entrada dos “pedidos” do projeto quanto mais informações puderem ser obtidas, melhor. Assim, foi criado o conceito de Meta Campo, que é um objeto visual configurado previamente e vinculado a determinado *projeto/categoria de solicitação, projeto, tarefa, risco, etc* e que é bastante útil para exibir campos adicionais em determinados formulários e com o formato visual pré-determinado (campo texto, caixa de seleção, calendário, text-area, etc). Os dados selecionados ou digitados nestes campos ficam armazenados na base de dados e podem ser utilizados em relatórios, KPIs, etc. A partir da versão 1.9.1, foi criado também o meta campo de “grid” que pode ser utilizado dentro do formulário com um componente mestre-detalle.

LIMITAÇÃO: A infra-estrutura que foi construída permite vincular Meta Campos para diversas entidades, porém atualmente apenas as telas de Solicitações, Tarefas, Faturas, Riscos e Projetos possuem esta funcionalidade. A partir da versão 1.3.0 foi incluído o conceito de Meta Formulário, onde os “meta campos” também são utilizados. Veja mais sobre isso no capítulo 28 deste documento. Outra limitação que em breve será resolvida é possibilitar a atribuição de permissões para cada meta campo. Isso seria bastante útil para definir se determinado usuário (baseado no seu papel no projeto) pode editar ou visualizar determinado meta campo do formulário.

Veja mais sobre os meta campos no site do projeto:

<http://sourceforge.net/apps/mediawiki/plandora/index.php?title=HowToCreateMetaFields>

Controle de Tarefas

9. **FUNCIONALIDADE:** O sistema deveria possibilitar ao recurso visualizar a sua “TO DO list” bem como possibilitar a visualização histórica de cada tarefa.

MOTIVAÇÃO: Da mesma forma que o cliente deve ter em mãos as informações histórias sobre as solicitações, é importante também disponibilizar o trâmite das tarefas realizadas pelo recurso. A grande motivação vinculada a esta funcionalidade é manter e enriquecer algo bastante negligenciado principalmente pelos projetos realizados por empresas de pequeno porte, mas que possui bastante valor agregado que é o *know-how* adquirido. É muito comum nos projetos, corrigir defeitos, implementar funcionalidades, quebrar tecnologias e todo o conhecimento fica na memória dos recursos humanos gerando problemas sérios de manutenibilidade.

10. *FUNCIONALIDADE: O recurso deveria ser capaz de reportar as horas utilizadas durante o ciclo de vida da tarefa e o líder deveria ser capaz de monitorar todo o trabalho executado pelos recursos do projeto.*

MOTIVAÇÃO: A grande maioria das empresas que se propõem a fazer planejamento de projetos utiliza normalmente ferramentas *standalone*. Basicamente, o gerente de projetos ou arquiteto monta o cronograma baseado em uma proposta e de tempos em tempos coleta com os recursos o status do que foi planejado. Em seguida é feito o replanejamento e o ciclo é reiniciado. Porém esta é uma tarefa mecânica que rouba um tempo precioso do gerente de projetos. Além disso, é muito comum o distanciamento da realidade em relação ao cronograma causando surpresas normalmente desagradáveis na fase final do projeto. Neste caso é importante que os apontamentos de horas trabalhadas sejam executados de forma automatizada. Além de ter um ganho de tempo, existe uma vantagem adicional que é o fato das tarefas estarem sempre espelhadas no diagrama de Gantt. No formato manual, é comum colocar tarefas "genéricas" no cronograma, para agilizar o processo de coleta, causando um efeito colateral de perda de controle no planejamento. Pelo modo automático, todas as tarefas realizadas e apontadas estarão no Gantt. Se alguma tarefa adicional aparecer devido a uma solicitação urgente, ela certamente aparecerá no diagrama de Gantt visualizado pelo líder.

11. *FUNCIONALIDADE: Uma das formas de monitoramento do trabalho executado pelos recursos deveria ser o diagrama de Gantt.*

MOTIVAÇÃO: O diagrama de Gantt é uma das formas tradicionais para visualizar a alocação de um grupo de recursos em um espaço de tempo. A motivação neste caso, surgiu da percepção de que em muitos projetos controlados com ferramentas *standalone* de controle de projetos, rapidamente o planejamento tornava-se defasado devido à dificuldade do gerente de projetos de atualizar constantemente o status das tarefas criando assim uma "realidade artificial" sobre a situação do projeto. A idéia era construir uma ferramenta que fosse atualizada de forma bidirecional, isto é, um recurso ao mudar o status de uma tarefa ou o tempo, estaria indiretamente alterando o Gantt visualizado pelo líder, já que a fonte dos dados é a mesma. Com isto o líder teria diariamente *feedback* do projeto sem mesmo ter cobrado dos recursos. Por outro lado, o líder poderia alterar livremente as tarefas ainda em aberto, através de manipulação no diagrama e ao salvar, alteraria automaticamente a configuração das "TO DO lists" dos recursos afetados.

LIMITAÇÕES: O Diagrama de Gantt ainda possui muito potencial a ser explorado. Existem muitas funcionalidades e idéias que ainda serão implementadas: Gráficos de Capacidade, Criação e Cancelamento de tarefas através do Gantt são algumas delas. Atualmente é possível apenas visualizar, redimensionar e mover tarefas existentes.

Veja mais sobre o Gantt Chart no site do projeto:

<http://sourceforge.net/apps/mediawiki/plandora/index.php?title=HowToReuseGantt>

12. *FUNCIONALIDADE: O sistema deveria guardar informação histórica sobre a relação: tempo estimado X tempo realizado*

MOTIVAÇÃO: Em muitos projetos, existem problemas com a diferença de percepção entre o gerente de projetos e o recurso que realiza a tarefa. Muitas vezes o gerente de projeto ou líder, estima determinado tempo para a execução de uma tarefa, mas por falta de controle, acaba não sabendo quão eficiente estão as suas estimativas. É fundamental numa ferramenta de controle de workflow, medir os dois tempos, pois isso fornece subsídios para os ajustes no planejamento. Além disso, auxilia o líder do projeto a medir a produtividade do recurso na medida em que ele pode comparar tarefas similares feitas por recursos diferentes e descobrir quais recursos executam melhor determinado tipo de tarefa e quais recursos necessitam de treinamento e em qual área.

LIMITAÇÕES: Algo bastante interessante, mas que ainda não entrou no *roadmap*, seria a utilização destes dados em gráficos desenhados no próprio Gantt. Mostrando o cronograma com as tarefas com os tempos estimados e as tarefas com os tempos realizados para que o líder tenha a percepção visual da diferença. Outra funcionalidade bastante interessante seria a criação de "congelamentos" de cronogramas. Isto é, em determinado momento o líder poderia criar uma Baseline do cronograma para utilizar posteriormente em comparações com o cenário corrente.

13. *FUNCIONALIDADE: Deveria ser possível atrelar custos aos recursos e incluir as receitas dos projetos*

MOTIVAÇÃO: Uma outra dimensão importante do planejamento de projetos, além do escopo e do tempo é sem dúvida nenhuma o custo. A idéia era, da mesma forma que é feito através de ferramentas como o *MSPProject*, atrelar valores de homem/hora aos recursos de um determinado projeto para facilitar a posterior análise de custos através de *KPIs* ou relatórios. Além disso, seria interessante que houvesse uma tela secundária onde os recursos pudessem fazer apontamentos de despesas de projetos e que só entrariam com despesas do projeto após a aprovação do líder, similar ao que a maioria das empresas faz com os reembolsos de despesas, porém utilizando para isso formulários standalone (word, excel, etc).

LIMITAÇÕES: Os dados referentes a custos limitam-se apenas aos custos relacionados de homem/hora por projeto. Não se tem ainda na ferramenta uma forma de vincular custos de projeto como operação, viagens, softwares, treinamentos, etc, bem como o controle de *Budgeting*, *Accounting* e *Charging* previstos nas boas práticas ITIL. A partir da versão 1.9.1 já é possível ao líder do projeto cadastrar e monitorar o ciclo de vida das faturas dos projetos.

14. **FUNCIONALIDADE:** *As informações de tarefas do sistema deveriam ser facilmente importadas ou exportadas.*

MOTIVAÇÃO: Os dados referentes aos projetos são fontes importantes tanto para análise quanto para utilização por outras ferramentas. Em muitos casos, o gerente do projeto possui grande familiaridade com ferramentas de mercado como *MSProject*, e é importante criar mecanismos para importar dados provenientes de ferramentas terceiras e também exportar os dados.

LIMITAÇÕES: Até a versão 1.9.1 as rotinas implementadas eram : importação de tarefas do *MSProject*, salvamento dos dados para o formato do Google Calendar e do *GanttProject*. Este último é um projeto *open source*, também hospedado no *SourceForge* e que almeja ser uma substituto *free* para o *MSProject*. Ele é escrito em Java e já possui versões estáveis. Está dentro do *roadmap*, a implementação de outras integrações como a exportação para o *MSProject* e o importação do *GanttProject*. Em relação à importação de dados, é importante ressaltar que ela será implementada apenas como fonte de dados para carga inicial. Existem alguns problemas nos processos de importação/exportação cíclicos, e por enquanto isso está fora do escopo do projeto, pelo menos até que se tenha uma forma genérica de viabilizar o processo cíclico independente da aplicação externa.

A partir da versão 1.3.0 foi criada a infra-estrutura na ferramenta para permitir que as classes Java de exportação fossem estendidas por classes customizadas. Assim, é possível criar de forma bem simples processos de exportação extremamente específicos utilizando os dados da ferramenta.

Já na versão 1.5.0, esta infra-estrutura foi expandida também para a importação de dados juntamente com a implementação da importação proveniente do *MSProject* versão 2003.

Aspectos Gerais

15. **FUNCIONALIDADE:** *O sistema deveria permitir a criação de Indicadores de Desempenho customizados por projeto.*

MOTIVAÇÃO: As empresas cada vez mais estão adotando as técnicas de *Balanced Scorecard* para definição de metas estratégicas. Em pequenas empresas de software, a maioria das metas relacionadas às perspectivas: financeira, cliente, processo e aprendizado, passam por uma ferramenta de workflow ou de planejamento de projetos, já que estes sistemas controlam o core do negócio. Assim, a motivação surgiu da oportunidade de fornecer essas informações através da criação de KPIs customizadas. Estas KPIs nada mais seriam do que consultas SQL na base de dados da ferramenta construídas segundo as necessidades de negócio específicas. A ferramenta faria apenas o papel de coletor destes indicadores e os mostraria em forma de tabelas na interface da ferramenta.

LIMITAÇÕES: Atualmente apenas os coletores diários estão implementados. A infra-estrutura montada permite a coleta de dados semanais, mensais, etc mais isso ainda não foi implementado.

Veja mais sobre a implementação de KPIs no site do projeto:
<http://sourceforge.net/apps/mediawiki/plandora/index.php?title=HowToCreateKPI>

16. *FUNCIONALIDADE: A ferramenta deveria ser leve, simples e web-based.*

MOTIVAÇÃO: Muitas ferramentas de workflow que foram analisadas durante a fase de levantamento de requisitos tinham um problema em comum que era a complexidade na utilização. A idéia era ter uma ferramenta onde tudo o que o usuário precisasse, estivesse intuitivamente agrupado, isto é, uma lista com as tarefas do usuário, as solicitações do usuário, os projetos, etc. Além disso, foi escolhida a implementação web, pois pelo fato de ser uma ferramenta de colaboração, a distribuição da ferramenta seria certamente um fator de complexidade, e com uma solução web isso seria minimizado. Além disso, a interface web facilitaria a utilização pelos clientes externos da empresa, esta sim, uma das grandes motivações do projeto.

17. *FUNCIONALIDADE: A ferramenta deveria ser multi-lingua*

MOTIVAÇÃO: Por ser uma ferramenta *open-source* hospedada em um portal utilizado mundialmente (*sourceforge.net*) era desejável que a ferramenta fosse portátil pelo menos em português, inglês e espanhol. Para isso, era necessário construir a ferramenta desde o início mantendo a infra-estrutura alinhada com este princípio, isto é, utilizar arquivos de recursos na camada de *front-end*, utilizar conversores de campos numéricos e datas para permitir a fácil formatação baseado em parâmetros de localização do browser.

18. *FUNCIONALIDADE: A ferramenta deveria ser de código aberto e livre de royalties*

MOTIVAÇÃO: Nunca foi intenção do projeto **Plandora** cobrar pela ferramenta. Primeiramente porque ela não implementa nenhuma idéia inédita. Ela pode ser sim uma opção barata que agrupa diversas funcionalidades numa ferramenta só, mas valorizar a cultura do software livre sempre foi um valor presente neste projeto. O projeto incentiva a reutilização de parte ou de todo o código-fonte implementado, e tenta sempre criar componentes mais desacoplados possíveis para permitir a sua reutilização (o componente de calendário, listas paginadas, o GanttChart, etc, são exemplos de elementos reutilizáveis).

19. *FUNCIONALIDADE: O sistema deveria permitir a criação de Relatórios por projeto totalmente customizados*

MOTIVAÇÃO: Cada projeto possui necessidades diferentes e cada gerente ou líder de projeto foca o controle do projeto por diferentes pontos de vista. Seguindo este princípio da customização, a funcionalidade de relatórios foi implementada possibilitando a criação de novos relatórios a partir de cláusulas SQL da mesma forma como é feito na criação das KPIs. Porém no caso do relatório, é necessário também gerar o layout, isto é, é necessário utilizar um editor para montar a “cara” do relatório (fontes, logos, posicionamento das colunas, paginação, gráficos, etc). Para isso, o **Plandora** utiliza os arquivos “.jasper” gerados pelo editor *iReport*. Esta ferramenta Java também é um projeto *open source* hospedada no *sourceForge* e é amplamente utilizada pelo mercado.

O **Plandora** também permite que se utilize palavras-chave dentro da SQL que são pré-processadas por ele antes de gerar o relatório no *iReport*. Isto possibilita a mudança dos dados de entrada pelo usuário que está gerando o relatório, tornando-o mais dinâmico.

Veja mais sobre relatórios através da demo no site do projeto:

<http://plandora.sourceforge.net/presentation/fluxoConfiguracao-Relatorio.htm>

20. *FUNCIONALIDADE: Criação de lista de riscos relacionada a solicitações, tarefa ou projetos.*

MOTIVAÇÃO: Uma das coisas mais interessantes dentro de um projeto de software e que na maioria das vezes é bastante negligenciada é a montagem de uma base de dados histórica com fins de redução de riscos para novos projetos. É muito comum realizar projetos e cometer os mesmos erros de projetos antigos porque as lições aprendidas em projetos anteriores não foram devidamente publicadas. No caso da lista de riscos idealizada para a ferramenta **Plandora**, o desafio é alinhar os conceitos de planejamento de riscos do PMI do ponto de vista de coleta de dados (qualificar e quantificar os riscos) e amarrá-los a determinadas entidades do sistema como, por exemplo, uma solicitação, tarefa ou mesmo ao projeto como um todo. Em seguida, utilizar essas informações nas pesquisas realizadas a partir da base de conhecimento dos projetos e possibilitar a armazenagem de informações sobre a ocorrência, mitigação etc caso o risco se materialize.

LIMITAÇÕES: Desde a versão 1.5.0 o gerenciamento de risco já é possível pela ferramenta, mas falta ainda criar um relatório nativo da ferramenta para exibí-los. De qualquer forma, os dados de risco são facilmente acessíveis através da pesquisa pela base de conhecimento (veja item 24) e se necessário pode-se criar relatórios customizáveis (veja item 19).

21. *FUNCIONALIDADE: Criação de alertas a eventos do sistema*

MOTIVAÇÃO: É muito importante que a ferramenta tenha mecanismos de pró-atividade a partir de critérios pré-definidos. Eventualmente é necessário que o sistema avise que determinados eventos estão ocorrendo, como por exemplo, *deadlines* críticos próximos, solicitações de determinados usuários ou prioridades diferenciadas, etc. Inicialmente o conceito de alerta poderia ser implementado especificamente através do envio de e-mails, mas a idéia é desenhar uma arquitetura que possibilite “plugar” facilmente outros tipos de canais (agentes) de saída.

Assim, desde a versão 1.0.0, a ferramenta possui os agentes nativos: e-mail, http, e log. A partir da versão 1.5.0 foi incluída também a verificação do conteúdo de arquivos. Porém a arquitetura foi montada para que os “agentes” da ferramenta pudessem ser estendidos (reimplementados ou sobrecarregados) para atender necessidades de negócios específicas.

Veja mais sobre a criação de agentes em:

<http://sourceforge.net/apps/mediawiki/plandora/index.php?title=HowToManageNotifications>

22. *FUNCIONALIDADE: Integração com ferramentas de controle de versão.*

MOTIVAÇÃO: A maioria das empresas que desenvolvem projetos de software utiliza ferramentas de controle de versão para facilitar a troca e o compartilhamento de arquivos entre os recursos do projeto de forma segura. As ferramentas de uso livre mais comuns atualmente são: CVS e Subversion. Basicamente o procedimento consiste em obter um arquivo do projeto do repositório, editá-lo e depois salvá-lo novamente no repositório para que outras pessoas possam utilizar a versão alterada do arquivo de projeto. Ao salvar o arquivo, o usuário pode opcionalmente escrever um comentário sobre a alteração que ele executou, e é esta informação que a ferramenta **Plandora** tem interesse em coletar, pois com isto, seria possível enriquecer consideravelmente (do ponto de vista qualitativo) a base histórica das tarefas na ferramenta. Além disso, outro benefício claro, é que com este tipo de procedimento seria possível saber para uma determinada solicitação, quais foram os arquivos de projeto que foram afetados pela alteração. Este tipo de informação seria a base para viabilizar um relatório de matriz de rastreabilidade, ou servir de base para um Change Manager fazer a análise de viabilidade de uma mudança.

LIMITAÇÕES: A partir da versão 1.9.1 foi disponibilizada a funcionada para vincular tarefas ou solicitações com itens de configuração do SVN. Estes links podem ser feitos tanto de forma ativa (via navegador de arquivos do repositório) quanto passivamente, através de configuração do *call-back* do SVN que antes de fazer o commit, requisita ao Plandora a permissão que libera ou não de acordo com as políticas de commit definidas pelo líder do projeto.

Veja mais sobre a integração do Plandora com o SVN em:

<http://sourceforge.net/apps/mediawiki/plandora/index.php?title=HowToIntegrateWithSVN>

23. *FUNCIONALIDADE: Implementação da funcionalidade de fórum de discussão.*

MOTIVAÇÃO: Seria interessante disponibilizar uma estrutura simples de fórum de discussão dentro do próprio **Plandora**. Embora a idéia pareça “reinventar a roda” já que existem fóruns consolidados na Internet e que são amplamente utilizados pela comunidade técnica, a idéia é criar algo não necessariamente com discussão técnica, mas focada na realidade do projeto ou da empresa onde o **Plandora** está instalado. Isto poderia ser, desde discutir onde será a próxima confraternização dos recursos projeto, até discussões estratégicas do projeto ou da empresa que não faria sentido discutir em fóruns públicos, com a vantagem de se ter toda a discussão “amarrada” a entidades da ferramenta como tarefas, solicitações, etc.

LIMITAÇÕES: Atualmente é possível apenas criar tópicos de discussão relacionados a uma solicitação específica. Esta funcionalidade pode ser habilitada e desabilitada para cada cliente / projeto. Porém este tipo de funcionalidade tem um grande potencial dentro da ferramenta que ainda será explorado.

24. *FUNCIONALIDADE: Implementação de uma ferramenta de busca para coletar dados da base de dados referentes aos projetos armazenados.*

MOTIVAÇÃO: Atualmente existe um *framework* de código livre dentro do projeto *Jakarta* chamado *Lucene* que é um poderoso *Engine* para indexação de conteúdo textual. A partir de um mecanismo dentro da ferramenta **Plandora** ocorre a indexação de informações preciosas como: conhecimentos adquiridos, riscos levantados, solicitações, histórico das tarefas, etc que são em seguida disponibilizados em uma tela de pesquisa nos moldes das ferramentas de busca amplamente utilizadas como *Google*, *Altavista*, etc.

Com isso, antes de realizar uma determinada tarefa, o recurso poderia fazer uso desta base de conhecimento para saber se determinado problema já foi resolvido anteriormente e por quem. Esta base de conhecimento poderia servir também para um *Problem Manager* fazer análise das solicitações fechadas (incidentes) em busca de causas raiz ainda desconhecidas. Este tipo de análise está em conformidade com as boas práticas ITIL.

25. *FUNCIONALIDADE: Implementação de uma estrutura para permitir a inter-relação entre tarefas.*

MOTIVAÇÃO: É muito comum nos projetos uma tarefa ter determinado nível de relacionamento com outra, por exemplo, uma determinada tarefa só pode começar se outra finalizar. Este tipo de mecanismo é a base do conceito de *work-flow*, onde é possível criar um grupo de tarefas que podem ser delegadas para diferentes recursos e que são

desempenhadas seguindo um fluxo coordenado (independente de ser sequencial ou em paralelo). Imaginando este conceito dentro do **Plandora**, é como se o líder de projeto pudesse criar não apenas tarefas isoladas para atender uma solicitação, mas um grupo de tarefas. Estas tarefas nada mais seriam do que as atividades de um processo de negócio previamente cadastrado (templates).

LIMITAÇÕES: A infra-estrutura para viabilizar o relacionamento entre as entidades já foi implementada. Nos formulários de Ocorrências de Projetos, Tarefas e Riscos já é possível criar vínculos entre as entidades e atribuir uma qualificação para o relacionamento. Um destes qualificadores é exatamente a possibilidade de bloquear o início de uma atividade até que outra seja concluída. A partir destes relacionamentos a ferramenta permite visualizar através de **mapas mentais** como as atividades se inter-relacionam. Este tipo de representação gráfica é útil para analisar o nível de rastreabilidade entre as solicitações, tarefas, stakeholders, etc. Porém ainda falta implementar a parte de criação de "templates" de tarefas, para tornar a ferramenta aderente ao conceito de *work-flow*.

26. **FUNCIONALIDADE:** Auto-encaminhamento de tarefas

MOTIVAÇÃO: Algumas atividades de projeto podem ser (dentro de um critério estabelecido pelo líder do projeto) delegadas automaticamente aos recursos sem nenhuma intervenção do líder. Isso ocorre tipicamente quando existe uma solicitação vinda de clientes críticos ou mesmo uma solicitação com prioridade alta, etc.

Assim, a idéia seria dar ao líder a possibilidade de criar no Plandora um cadastro de políticas e filtros para encaminhamento automático dentro do projeto.

LIMITAÇÕES: Funcionalidade ainda não implementada.

27. **FUNCIONALIDADE:** Recurso coringa

MOTIVAÇÃO: Tipicamente, o processo de alocação de tarefas pelo recurso se dá através da análise do perfil do recurso X o escopo da tarefa bem como a análise da alocação dos recursos, urgência, skills, etc. Porém, muitas vezes, surgem tarefas que podem ser executadas por qualquer recurso, e assim o líder fica em dúvida de qual recurso escolher. Para estes casos, a idéia seria implementar no Plandora o conceito de Recurso Coringa, onde no momento da alocação o líder delegaria a tarefa não para um recurso específico, mas para um recurso "Anyone". A partir daí, esta atividade apareceria automaticamente na *TO DO list* de todos os recursos do projeto. O primeiro que puxa-se a tarefa para si, ganharia a alocação e a tarefa sairia da lista dos demais.

LIMITAÇÕES: Esta funcionalidade está disponível a partir da versão 1.5.0. Mas ela pode ainda ser melhorada. Por exemplo, hoje é possível

que o líder selecione para cada projeto, as funções exercidas pelo recurso no projeto (testador, DBA, gerente, etc). Seria interessante que além do encaminhamento para "Anyone" fosse possível o encaminhamento para um determinado papel, seguindo a mesma lógica do recurso "coringa". Outra funcionalidade desejável seria a criação de um botão na tela principal para exibição deste Banco de Tarefas "sem dono". Isto seria útil para não poluir tanto a TO DO list do recurso.

28. *FUNCIONALIDADE: Meta Formulários*

MOTIVAÇÃO: Complementando a funcionalidade dos meta campos, que são elementos que podem ser incluídos em alguns formulários da ferramenta (solicitações, projetos, etc), uma outra maneira de customizar ainda mais a ferramenta são os Meta Formulários que são formulários estanque criados através de cadastro na base de dados e que contém um conjunto de meta campos. Este tipo de formulários é útil caso se queira colocar alguma lógica de negócio específica dentro da ferramenta e utilizar os dados cadastrados nestes formulários em métricas ou relatórios na própria ferramenta. Os meta formulários são sempre criados pelo usuário root. E ele também que pode atribuir um determinado meta formulário para quaisquer usuários cadastrados na ferramenta. A partir daí, este usuário será capaz de visualizar o formulário a partir do link "Opções" e incluir, remover ou atualizar dados como se fosse um formulário nativo da ferramenta.

LIMITAÇÕES: Falta ainda implementar algumas funcionalidades importantes de usabilidade, por exemplo filtros de pesquisa. A partir da versão 1.5.0 foi implementada a possibilidade de cadastro de código JavaScript para ser executado pela ferramenta em momentos específicos do ciclo de vida do meta formulário como por exemplo: ao carregar, antes de salvar, etc.

29. *FUNCIONALIDADE: Repositório de Arquivos*

MOTIVAÇÃO: Dentro de um projeto, é bastante comum o uso de ferramentas de controle de versão, para armazenamento do código fonte e dos documentos do projeto. A idéia seria utilizar o Plandora como um "client" desta ferramenta, mas não necessariamente para executar *commits* de arquivos. O interesse do Plandora está em ter acesso e indexar a base de arquivos do repositório para centralizar todo o conhecimento do projeto em uma base única e com pesquisa "google-like". Na verdade o líder do projeto seria o responsável, através de uma interface gráfica no Plandora, por selecionar quais arquivos ou pastas do repositório seriam "indexáveis". Uma segunda funcionalidade seria o líder do projeto atribuir para determinados arquivos do repositório qual o papel mínimo para visualização. Com isso, seria possível deixar sempre disponível ao cliente final do projeto, determinados arquivos (documentos entregáveis) que ele teria acesso

via a interface web da ferramenta, evitando a necessidade de envio de versões de arquivos por e-mail.

LIMITAÇÕES: A partir da versão 1.5.0, já é possível a navegação e o download de arquivos do repositório para os usuários com papel de líder ou recurso. Ainda não foi implementado o conceito de indexação e nem a atribuição de papel mínimo para visualização. Além disso, atualmente, a funcionalidade existe apenas para o Subversion (SVN).

30. *FUNCIONALIDADE: Gráficos de Métricas de Projeto*

MOTIVAÇÃO: Embora muitos indicadores possam ser exibidos via relatório, uma ferramenta bastante útil seria poder visualizar a “saúde” de projeto de forma gráfica na tela principal do Plandora. Muitas ferramentas de gerenciamento de projeto chamam isso de “Dashboard”. O Plandora possui o conceito de *Gadget*, que são gráficos que podem ser selecionados tanto pelos líderes quanto pelos recursos a partir do formulário de “Opções” e que são visualizados sempre na parte direita da tela principal. Na arquitetura planejada dentro do Plandora para os Gadgets foi contemplada a possibilidade de se “plugar” novos gráficos a partir das classes Java de interface disponíveis no pacote de instalação do projeto (*plandoraInterface.jar*). Assim, pode-se criar gráficos customizados para serem exibidos no *dashboard* lateral da ferramenta.

31. *FUNCIONALIDADE: Pesquisas (surveys)*

MOTIVAÇÃO: É bastante recorrente em projetos de software a necessidade da geração de questionários para ser utilizado no levantamento de requisitos, em pesquisas de satisfação, análise de gap ou mesmo para coletar feedback de determinado evento do projeto, por exemplo um treinamento. Seria interessante que o sistema tivesse uma estrutura para que o líder do projeto pudesse criar pesquisas customizadas para cada projeto e que houvesse a opção de criar pesquisas anônimas caso o líder achasse conveniente.

Tabela relacionando FUNCIONALIDADE X Versão da ferramenta

[illegible]

Tabela relacionando FUNCIONALIDADE X Perfil de Usuário

	Requisito	Cliente	Recurso	Líder	Root
1	SPOC	✓	✓	✓	
2	Histórico de Solicitações	✓	✓	✓	
3	Multiprojetos	✓	✓	✓	✓
4	Hierarquia Projetos	✓	✓	✓	✓
5	Papéis de Usuários			✓	✓
6	Aceite/Rejeição Solicitações			✓	
7	Definição Data de Entrega			✓	
8	Meta Campos	uso	uso	uso	✓
9	To Do List		✓	✓	
10	Apontamento de Horas		✓	✓	
11	Gant Chart		✓	✓	
12	Tempo Estim. X Tempo Real		✓	✓	
13	Custo Recursos Humanos			✓	✓
14	Import / Export		✓	✓	
15	KPI			uso	✓
16	Web-based	✓	✓	✓	✓
17	Internacionalização	✓	✓	✓	✓
18	Open-Source/Freeware	✓	✓	✓	✓
19	Relatórios		uso	uso	✓
20	Lista de Riscos			✓	
21	Alertas	uso	uso	uso	✓
22	Integração c/ Controle de Versão		uso	uso	
23	Fórum de Discussão	✓	✓	✓	
24	Ferramenta de Busca		✓	✓	
25	Relacionamento Atividades		✓	✓	
26	Auto-encaminhamento				
27	Recurso Coringa				
28	Meta Formulários	uso	uso	uso	✓
29	Repositório de Arquivos	uso	✓	✓	
30	Gráficos e Métricas do Projeto		✓	✓	
31	Form de Pesquisas (surveys)	uso	uso	✓	

Para Conhecer Mais...

Entre no *site* do projeto, veja os *screenshots* e *demonstrações*. Aprenda a instalar a ferramenta, a customizar algumas funcionalidades e acompanhe os artigos sobre as novas funcionalidades e tendências.

<http://www.plandora.org>