**N2 – Engenharia de Software**

**Nomes:**

Gustavo Antonio Ferreira – 081160042

Maria de Fátima Rocha Silveira – 081160042

Matheus Augusto Maria – 081160017b

Faculdade de Tecnologia Termomecanica

2019

**Plano de iterações - Concepção**

# **Introdução**

## **Propósito**

O propósito deste documento é reunir todas as informações necessárias para iniciar o desenvolvimento de um protótipo de software de fluxo de entrada de pedidos e saída de produtos. Durante essa iteração os requerimentos do sistema serão definidos e um plano de execução de alto nível de todo o projeto deve ser elaborado. Essa primeira iteração irá conduzir as análises dos casos de negócio para prosseguimento do projeto.

## **Escopo**

Este documento desenvolvido se aplica ao desenvolvimento de um software com fluxo de entrada de pedidos e saída de produtos. O gerente de projeto e a equipe de desenvolvimento se baseará nesse documento ao longo do processo de desenvolvimento do sistema.

## **Plano**

Essa iteração inicial esclarecerá os requerimentos do projeto e estabelecerá os casos de negócio para o sistema de pedidos. Ao final desta iteração será decidido qual funcionalidade será priorizada no desenvolvimento e por qual caminho o software será desenvolvido como um todo.

|  |  |  |  |
| --- | --- | --- | --- |
| **Atividade** | **Inicio** | **Fim** | **Duração** |
| Concepção | Imediato após pedido do cliente | 21 dias após o pedido | 21 dias |
| Fase inicial de concepção | Imediato após pedido do cliente | 2 dias após início da concepção | 2 dias |
| Definir ferramentas de desenvolvimento (criação de tarefas); Definir controle de mudanças; | Após término da atividade inicial de concepção | 3 dias após início da concepção | 1 dia |
| Revisão das atividades descritas | Após término da atividade de definição de ferramentas e controle de mudanças | 4 dias após início da concepção | 1 dia |
| Reunião de inicial (definições básicas) | Após término da atividade de revisão das atividades descritas | 5 dias após início da concepção | 1 dia |
| Desenvolvimento dos entregáveis | Após término da atividade inicial da concepção do projeto | 13 dias após início da concepção | 8 dias |
| Coleta de requisitos | Imediato após reunião inicial | 6 dias após início da concepção do projeto | 1 dia |
| Visão do projeto criada e revisada | Imediato após reunião inicial | 7 dias após início da concepção | 1 dia |
| Desenvolvimento dos casos de uso preliminares e inseridos no controle de revisão | Após término da atividade de criar visão do projeto | 9 dias após início da concepção | 2 dias |
| Pesquisa de caso de uso criada e revisada | Após desenvolvimento dos casos de uso preliminares | 10 dias após início da concepção | 1 dia |
| Especificações suplementares criadas e revisadas; | Após pesquisa de casos de uso | 11 dias após início da concepção | 1 dia |
| Caso de negócio criado e revisado; | Após pesquisa de casos de uso | 11 dias após início da concepção | 1 dia |
| Vocabulário do projeto coletado e revisado; | Após pesquisa de casos de uso | 11 dias após início da concepção | 1 dia |
| Design criado e revisado | Após reunião inicial da concepção | 6 dias após início da concepção | 1 dia |
| Mockup da interface com o usuário criada e revisada | Após reunião inicial da concepção | 6 dias após início da concepção | 1 dia |
| Plano de transferência de conteúdo e programação criada e revisada | Após coleta de vocabulário do projeto | 12 dias após início da concepção | 1 dia |
| Validar se o projeto vai atender as necessidades | Após plano de transferência de conteúdo criado e revisado | 13 dias após início da concepção | 1 dia |
| Finalização do entregáveis | Após validação e assinatura do projeto | 13 dias após início da concepção | 1 dia |
| Aplicação dos objetos desenvolvidos na concepção | Após finalização dos entregáveis | 13 dias após início da concepção | 8 dias |
| Reunião de verificação de qualidade | Após finalização dos entregáveis | 13 dias após início da concepção | 1 dia |
| Realizar garantia da qualidade | Após reunião de verificação de qualidade | 14 dias após início da concepção | 1 dia |
| Primeiras estimativas criadas e revisadas | Após realizar a garantia da qualidade | 16 dias após início da concepção | 2 dias |
| Plano de entregas iterativos criado e revisado | Após primeiras estimativas criadas e revisadas | 17 dias após início da concepção | 1 dia |
| Criar proposta para a fase de elaboração | Após finalização dos entregáveis | 14 dias após início da concepção | 1 dia |
| Finalização dos documentos para checkpoint da concepção | Após finalização dos planos de entregas criados e revisados | 18 dias após início da concepção | 1 dia |
| Entrega da proposta para fase de elaboração | Após finalização dos documentos para checkpoint | 19 dias após início da concepção | 1 dia |
| Checkpoint: Revisão dos documentos como um todo e decisão final se o projeto terá continuidade | Após entrega da proposta para fase de elaboração | 21 dias após início da concepção | 2 dias |
| Conclusão da concepção do projeto | Após checkpoint | 21 dias após início da concepção |  |

Serão gerados os seguintes entregáveis nesta fase.

|  |  |  |
| --- | --- | --- |
| **Artefato** | **Entregável** | **Responsável** |
| Modelo de negócio | Glossário | Analista de negócios |
| Conjunto de requerimentos | Visão do projeto | Analista de negócios |
| Especificações dos casos de uso | Analista de negócios Especialista de requerimentos |
| Especificações suplementares | Arquiteto |
| Modelo de caso de uso | Arquiteto |
| Gerenciamento | Plano de iterações preliminar | Gerente de projeto |
| Plano do projeto | Gerente de projeto |
| Cronograma do processo | Gerente de projeto |
| Lista de riscos | Gerente de projeto |
| Avaliação das iterações preliminares | Gerente de projeto |
| Plano de gerenciamento de configuração | Gerente de projeto Especialista de requerimentos |
| Guias | Configuração de ambiente | Especialista de requerimentos |

## **Critério de avaliação**

O objetivo da iteração preliminar é definir o sistema com o nível necessário de detalhe para que possa ser feito um julgamento sólido sobre a viabilidade do projeto.  
Ao final da iteração, uma revisão do caso de negócio definirá se o projeto terá continuidade ou se não será viável a implementação do mesmo.

**Plano de iterações - Construção**

# **Introdução**

## **Propósito**

Este plano de iteração descreve detalhadamente os planos para a fase de construção. Durante essa iteração apenas os quatro casos de uso mais importantes serão desenvolvidos.

## **Escopo**

Este documento desenvolvido se aplica a fase de construção do desenvolvimento de um software com fluxo de entrada de pedidos e saída de produtos. O gerente de projeto e a equipe de desenvolvimento se baseará nesse documento ao longo do processo de desenvolvimento do sistema**.**

## **Plano**

A fase de construção irá de fato implementar os quatro casos de uso de maior importância para o projeto.

|  |  |  |  |
| --- | --- | --- | --- |
| **Atividade** | **Inicio** | **Fim** | **Duração** |
| Fase inicial de construção | Após fase de elaboração | 30 dias após término da concepção | 23 dias |
| Reunir toda a equipe e fazer reunião para início da fase de construção | Após fase de elaboração | 1 dia após início da construção | 1 dia |
| Revisões técnicas | Após reunião de início da fase de construção | 2 dias após início da construção | 1 dia |
| Plano e casos de testes para próxima iteração criados e revisados | Após reunião de início da fase de construção | 2 dias após início da construção | 1 dia |
| Plano de integração criado e revisado | Após reunião de início da fase de construção | 2 dias após início da construção | 1 dia |
| Descrição do plano de construção para esta iteração, incluindo milestones, criado e revisado | Após reunião de início da fase de construção | 2 dias após início da construção | 1 dia |
| Atualização do plano de testes e refinamento dos casos de testes com foco na iteração atual | Após reunião de início da fase de construção | 1 dias após início da construção do projeto | 1 dia |
| Criar ambiente de teste e uma build do projeto para iteração atual | Após reunião de início da fase de construção | 1 dias após início da construção | 1 dia |
| Codificação do caso de uso de cadastro de funcionários | - | - | - |
| Refinamento da arquitetura e do banco de dados com foco na iteração atual | Após criar ambiente de teste e uma build do projeto para iteração atual | 2 dias após início da construção | 1 dia |
| Desenvolver código, pesquisa por bugs e ferramentas de desenvolvimento | Após refinamento da arquitetura e do banco de dados | 6 dias após início da construção | 4 dias |
| Teste unitário | Após desenvolver código, pesquisa por bugs e ferramentas de desenvolvimento | 7 dias após início da construção | 1 dia |
| Codificação do caso de uso de cadastro de produtos | - | - | - |
| Refinamento da arquitetura e do banco de dados com foco na iteração atual | Após Teste unitário da iteração passada | 2 dias após início da construção | 1 dia |
| Desenvolver código, pesquisa por bugs e ferramentas de desenvolvimento | Após refinamento da arquitetura e do banco de dados | 6 dias após início da construção | 4 dias |
| Teste unitário | Após desenvolver código, pesquisa por bugs e ferramentas de desenvolvimento | 7 dias após início da construção | 1 dia |
| Codificação do caso de uso de recebimento de pedidos | - | - | - |
| Refinamento da arquitetura e do banco de dados com foco na iteração atual | Após Teste unitário da iteração passada | 8 dias após início da construção | 1 dia |
| Desenvolver código, pesquisa por bugs e ferramentas de desenvolvimento | Após refinamento da arquitetura e do banco de dados | 12 dias após início da construção | 4 dias |
| Teste unitário | Após desenvolver código, pesquisa por bugs e ferramentas de desenvolvimento | 13 dias após início da construção | 1 dia |
| Codificação do caso de uso de pagamento de pedido | - | - | - |
| Refinamento da arquitetura e do banco de dados com foco na iteração atual | Após Teste unitário da iteração passada | 8 dias após início da construção | 1 dia |
| Desenvolver código, pesquisa por bugs e ferramentas de desenvolvimento | Após refinamento da arquitetura e do banco de dados | 12 dias após início da construção | 4 dias |
| Teste unitário | Após desenvolver código, pesquisa por bugs e ferramentas de desenvolvimento | 13 dias após início da construção | 1 dia |
| Teste de integração | Após Teste unitário da iteração passada | 15 dias após início da construção | 2 dias |
| Teste de regressão | Após revisões técnicas | 16 dias após início da construção | 1 dia |
| Integração com o sistema | Após Teste de integração | 16 dias após início da construção | 1 dia |
| Documentar problemas | Após integração com o sistema | 17 dias após início da construção | 1 dia |
| Atualização em manuais de usuário e obter aprovações | Após Documentar problemas | 18 dias após início da construção | 1 dia |
| Instalação para aceitação do usuário | Após atualização em manuais de usuário e obter aprovações | 19 dias após início da construção | 1 dia |
| Checkpoint: reunião de apresentação do software ao usuário | Após Instalação para aceitação do usuário | 20 dias após início da construção | 1 dia |
| Teste de aceitação do usuário | Após checkpoint | 21 dias após início da construção | 1 dia |
| Release | Após Teste de aceitação do usuário | 22 dias após início da construção | 1 dia |
| Reunião para garantia da qualidade | Após checkpoint | 22 dias após início da elaboração | 1 dia |
| Aplicar informações coletadas em reunião | Após reunião para garantia da qualidade | 23 dias após início da elaboração | 1 dia |
| Revisão geral de documentos | Após reunião para garantia da qualidade | 23 dias após início da elaboração | 1 dia |
| Fase de construção completa | Após revisão geral de documentos | 23 dias após início da elaboração | 1 dia |

Serão gerados os seguintes entregáveis nesta fase.

|  |  |  |
| --- | --- | --- |
| **Artefato** | **Entregável** | **Responsável** |
| Gerenciamento | Plano de iterações | Gerente de projeto |
| Avaliação do status do projeto | Gerente de projeto |
| Avaliação das iterações | Gerente de projeto |
| Plano de testes | Gerente de projeto |
| Modelagem | Modelo implementado | Arquiteto |
| Implementação | Plano de integração | Gerente de projeto |
| Programas de testes, scripts | Testador |
| Procedimento de testes | Testador |
| Relatório de avaliação dos testes | Gerente de projeto |
| Componentes | Desenvolvedor |

## **Critério de avaliação**

O objetivo principal é implementar com sucesso todos os casos de uso selecionados para essa iteração.  
Cada documento entregável será revisado e estará sujeito a aprovação por toda a equipe.  
Planos de teste formais servirão de base para avaliação do sucesso dessa iteração.

**Plano de iterações - Elaboração**

# **Introdução**

## **Propósito**

Este documento descreve detalhadamente os planos para esta fase de elaboração. Durante essa fase será definida a arquitetura do sistema e o plano de execução de alto nível do projeto será refinado.

## **Escopo**

Este documento desenvolvido se aplica ao desenvolvimento de um software com fluxo de entrada de pedidos e saída de produtos. O gerente de projeto e a equipe de desenvolvimento se baseará nesse documento ao longo do processo de desenvolvimento do sistema.

## **Plano**

A fase de elaboração irá terminar a análise dos requisitos, além disso todos os casos de usos serão modelados e analisados. Por final um protótipo da arquitetura será desenvolvido para provar a viabilidade e performance da arquitetura para o primeiro release.

|  |  |  |  |
| --- | --- | --- | --- |
| **Atividade** | **Inicio** | **Fim** | **Duração** |
| Elaboração | Após fase de concepção | 30 dias após término da concepção | 30 dias |
| Fase inicial de elaboração | Após fase de concepção | 1 dias após início da elaboração | 1 dia |
| Definir ferramentas de desenvolvimento (criação de tarefas); | Após término da atividade inicial de elaboração | 3 dias após início da elaboração | 2 dias |
| Planejamento da reunião para início da elaboração | Após término da atividade de definição de ferramentas | 5 dias após início da elaboração | 2 dias |
| Reunião de início | Após término da atividade de Planejamento da reunião para início da elaboração | 6 dias após início da elaboração | 1 dia |
| Plano para garantia de qualidade, criado e revisado. | Após planejamento da reunião para início da elaboração | 6 dias após início da elaboração | 1 dia |
| Protótipos de interface com o usuário e padrões de estilo | Após término da reunião inicial | 9 dias após início da elaboração do projeto | 3 dias |
| Componentes Web criados (agilidade no desenvolvimento) | Após término da reunião inicial | 7 dias após início da elaboração | 1 dia |
| Protótipo de interface com usuário inicial criado e revisado | Após término da reunião inicial | 7 dias após início da elaboração | 1 dia |
| Protótipo inicial de estilos criado e revisado | Após término da reunião inicial | 7 dias após início da elaboração | 1 dia |
| Atualização do backlog do projeto | Após término da reunião inicial | 9 dias após início da elaboração | 3 dias |
| Finalização dos protótipos de interface e estilos | Após atualização do backlog do projeto | 10 dias após início da elaboração | 1 dia |
| Coleta de requisitos para documentação e manuais de treinamento | Após finalização dos protótipos de interface e estilos | 12 dias após início da elaboração | 1 dia |
| Esboço da documentação e manuais de treinamento | Após coleta de requisitos para documentação e manuais de treinamento | 13 dias após início da elaboração | 1 dia |
| Detalhar requisitos de desenvolvimento | Após esboço da documentação e manuais de treinamento | 15 dias após início da elaboração | 2 dias |
| Completar elaboração do modelo de caso de uso | Após esboço da documentação e manuais de treinamento | 14 dias após início da elaboração | 1 dia |
| Especificações suplementares | Após esboço da documentação e manuais de treinamento | 14 dias após início da elaboração | 1 dia |
| Reavaliação dos documentos gerados na fase anterior (Vocabulário, plano de conteúdo, ...) | Após esboço da documentação e manuais de treinamento | 14 dias após início da elaboração | 1 dia |
| Detalhamento do protótipo de interface | Após esboço da documentação e manuais de treinamento | 14 dias após início da elaboração | 1 dia |
| Manuais de utilização criados e revisados | Após especificações suplementares | 15 dias após início da elaboração | 1 dia |
| Revisão de documentos (Plano de entregas iterativos, plano de desenvolvimento, caso de caso de negócio) | Após manuais de utilização criados e revisados | 16 dias após início da elaboração | 1 dia |
| Checkpoint da elaboração |  |  |  |
| Preparação e execução da reunião de checkpoint de requerimentos da elaboração | Revisão de documentos (Plano de entregas iterativos, plano de desenvolvimento, caso de caso de negócio) | 18 dias após início da elaboração | 2 dias |
| Fazer com que a equipe inteira finalize as tarefas para release | Após Preparação e execução da reunião de checkpoint de requerimentos da elaboração | 19 dias após início da elaboração | 1 dia |
| Revisão dos documentos como um todo e decisão final se o projeto terá continuidade para checkpoint | Após Preparação e execução da reunião de checkpoint de requerimentos da elaboração | 19 dias após início da elaboração | 1 dia |
| Revisões técnicas | Após checkpoint | 20 dias após início da elaboração | 1 dia |
| Definir convenções do modelo | Após revisões técnicas | 21 dias após início da elaboração | 1 dia |
| Identificar conceitos chave | Após definir convenções do modelo | 22 dias após início da elaboração | 1 dia |
| Criar execuções dos casos de uso | Após Identificar conceitos chave | 22 dias após início da elaboração | 1 dia |
| Revisão geral dos resultados | Após execuções dos casos de uso | 24 dias após início da elaboração | 2 dias |
| Análise geral da arquitetura | Após revisão geral dos resultados | 26 dias após início da elaboração | 2 dias |
| Checkpoint: Arquitetura modelada | Após análise geral da arquitetura | 27 dias após início da elaboração | 1 dia |
| Fase final da elaboração | Após checkpoint | 29 dias após início da elaboração | 2 dias |
| Reunião para garantia da qualidade | Após checkpoint | 28 dias após início da elaboração | 1 dia |
| Aplicar informações coletadas em reunião | Após reunião para garantia da qualidade | 29 dias após início da elaboração | 1 dia |
| Revisão geral de documentos | Após reunião para garantia da qualidade | 29 dias após início da elaboração | 1 dia |
| Fase de elaboração completa | Após revisão geral de documentos | 30 dias após início da elaboração | 1 dia |

Serão gerados os seguintes entregáveis nesta fase.

|  |  |  |
| --- | --- | --- |
| **Artefato** | **Entregável** | **Responsável** |
| Conjunto de requerimentos | Protótipo de interface com usuário | Designer de telas |
| Gerenciamento | Plano de iterações | Gerente de projeto |
| Avaliação do status do projeto | Gerente de projeto |
| Avaliação das iterações | Gerente de projeto |
| Plano de testes | Gerente de projeto |
| Guias | Guias de modelo | Arquiteto |
| Guias para usuário | Designer de telas |
| Modelagem | Análise do modelo | Arquiteto |
| Modelo | Arquiteto |
| Documento da arquitetura do software | Arquiteto |
| Implementação | Plano de integração | Gerente de projeto |
| Protótipo de arquitetura (base do projeto) | Arquiteto |
| Programas de testes, scripts | Testador |
| Procedimento de testes | Testador |
| Relatório de avaliação dos testes | Gerente de projeto |

## **Critério de avaliação**

O objetivo principal é completar a análise e modelagem dos casos de usos selecionados anteriormente e partir deles desenvolver um protótipo funcional da arquitetura do software.  
A associação de riscos relacionadas a modelagem da arquitetura ou performance será realizada ao fim desta fase.  
Cada documento entregável será revisado e estará sujeito a aprovação por toda a equipe.

**Plano de iterações - Transição**

# **Introdução**

## **Propósito**

Este plano de iteração descreve detalhadamente os planos para a fase de transição. Essa fase tem como objetivo especificar os planos de manutenção e suporte do software desenvolvido.

## **Escopo**

Este documento desenvolvido se aplica a fase de transição do desenvolvimento de um software com fluxo de entrada de pedidos e saída de produtos. O gerente de projeto e a equipe de desenvolvimento se baseará nesse documento ao longo do processo de desenvolvimento do sistema.

## **Plano**

A fase de transição irá preparar a equipe para prestar suporte e manter o bom funcionamento do sistema desenvolvido.

|  |  |  |  |
| --- | --- | --- | --- |
| **Atividade** | **Inicio** | **Fim** | **Duração** |
| Reunião inicial da fase de transição | Após fase de construção | 9 dias após início da transição | 9 dias |
| Compartilhar procedimentos, checklist revisado | Após reunião de início da fase de transição | 1 dias após início da transição | 1 dia |
| Compartilhar plano de contingência e disaster recovery, revisado | Após reunião de início da fase de transição | 1 dias após início da transição | 1 dia |
| Treinar equipes | Após reunião de início da fase de transição | 1 dias após início da transição | 1 dia |
| Preparar ambiente para suporte, internet, dados, integração com legado. | Após treinar equipes | 2 dias após início da transição | 1 dia |
| Teste de funcionalidade | Após preparar ambiente para suporte, internet, dados, integração com legado. | 3 dias após início da transição do projeto | 1 dia |
| Teste de performance | Após teste de funcionalidade | 4 dias após início da transição | 1 dia |
| Teste de inicialização | Após teste de performance | 5 dias após início da transição | 1 dia |
| Leves modificações para melhoria de performance | Após teste de inicialização | 6 dias após início da transição | 1 dia |
| Teste de aceitação do usuário | Após leves modificações para melhoria de performance | 7 dias após início da transição | 1 dia |
| Treinamento para área de negócios | Após teste de aceitação do usuário | 8 dias após início da transição | 1 dia |
| Revisão dos manuais de operação | Após teste de aceitação do usuário | 8 dias após início da transição | 1 dia |
| Reunião para garantia da qualidade | Após revisão dos manuais de operação | 9 dias após início da transição | 1 dia |
| Revisão do checklist e verificar se tudo foi executado | Após reunião para garantia da qualidade | 9 dias após início da transição | 1 dia |
| Fase de construção completa | Após revisão geral de documentos | 9 dias após início da transição | 1 dia |

Serão gerados os seguintes entregáveis nesta fase.

|  |  |  |
| --- | --- | --- |
| **Artefato** | **Entregável** | **Responsável** |
| Gerenciamento | Plano de iterações | Gerente de projeto |
| Avaliação do status do projeto | Gerente de projeto |
| Avaliação das iterações | Gerente de projeto |
| Implementação | Relatório de avaliação dos testes | Gerente de projeto |

## **Critério de avaliação**

O objetivo principal é tornar a equipe capaz de prestar um bom serviço e manter o software em funcionamento.  
O sucesso será mensurado através da aceitação do usuário final e avaliação de operações.

**Plano de Práticas do XP – 3 práticas adotadas**

1. **Teste**

Os testes serão utilizados pois com eles os programadores conseguem saber exatamente o que o sistema deve e não deve fazer e, além disso, como são feitos antes da programação, também funcionam como especificação do projeto. Nos testes o cliente possui uma participação ativa pois é a partir das informações passadas por ele que os testes serão elaborados.

Os testes são do tipo TDD, Test-Driven Design, onde os testes são elaborados antes da construção do software e o desenvolvedor programa e executa os testes quase que simultaneamente, eliminando aquele processo de desenvolvimento onde primeiro se programa e depois o programador passa para outro membro da equipe testar.

Os TDDs são testes automatizados, o que diminui o trabalho de “testar na mão” todas as funcionalidades, fazendo com que o programador consiga executar os testes a qualquer momento para testar todas as funcionalidades do sistema e checar se a mudança que está realizando gerou um bug ou não. Também serão utilizados os testes unitários que são uma forma de testes automatizados, porém testam apenas classes e não funcionalidades por inteiro.

1. **Propriedade Coletiva**

A Propriedade Coletiva consiste em disseminar conhecimento do código à toda equipe, evitando que apenas um programador conheça determinada parte do software.

O princípio de Propriedade Coletiva será utilizado como um artefato de proteção ao projeto pois com ele é possível garantir que todos os programadores conheçam todo o sistema e são capazes de atuar em demandas de qualquer parte do software. Assim, é possível evitar grandes transtornos que podem ser causados com eventuais saídas/entradas de programadores na equipe.

1. **Reunião diária**

A reunião diária é imprescindível para o ecossistema da equipe do projeto. Nela os membros do projeto se reúnem diariamente por um curto período de tempo e compartilham as atividades que estão executando, seus impedimentos, conquistas e preocupações. Assim todos da equipe ficam cientes das atividades e progresso de todos, o que evita grandes surpresas no projeto.

Com as reuniões diárias é possível ter uma visão melhor do andamento do projeto e identificar necessidades como priorizações, mudanças no escopo, necessidade de contratações de novos funcionários e etc.

**Avaliação dos itens de acordo com Manifesto Ágil**

1. **Indivíduos e interações em vez de processos e ferramentas**

Todo software é construído por pessoas e, portanto, o indivíduo deve ser valorizado. É necessário construir projetos baseados em indivíduos com ambiente, suporte e condições de trabalho adequadas.

A confiança é outro item importantíssimo. Os membros da equipe devem se sentir seguros, capazes e amparados ao fazer parte do desenvolvimento do sistema. Os processos e ferramentas são bem-vindos, porém não é ideal que sejam muito burocráticos, o ideal é que sejam simples e úteis. Portanto todos os processos e ferramentas devem ser constantemente revisados e sempre que possível, simplificados.

1. **Software funcional em vez de documentação abrangente**

É natural que exista a necessidade de uma documentação do projeto, porém ela deve ser simples e a equipe não deve gastar muita energia no processo de documentação pois, o que realmente importa para o cliente é o software em si, e um software funcional é um software funcionando concretamente, que satisfaz o cliente, com entregas constantes e valor agregado.

1. **Colaboração do cliente em vez de negociação do contrato**

Colaboração é a palavra chave para o relacionamento com o cliente, ambos, empresa contratada e contratante devem trabalhar em conjunto, um sempre visando o progresso do outro. Esse tipo de relação é muito saudável e intensifica a confiança entre as empresas, pois ambos trabalham como uma única equipe com um objetivo. Os contratos devem existir, porém não devem ser impeditivos para ajudar os clientes a conseguir cada vez mais vantagens competitivas.

1. **Respostas às mudanças em vez de seguir um plano**

Ter um plano é importante, porém ele deve ser flexível e ir se adaptando de acordo com as necessidades do cliente. A cultura de que mudanças, mesmo que pedidas tardiamente no desenvolvimento do projeto, são ruins deve ser eliminada. As mudanças são importantes e são oportunidades de fazer com que sua própria empresa e a do cliente se desenvolva cada vez mais. Portanto os projetos devem ser flexíveis e capazes de aceitar realizar mudanças no escopo do projeto, sempre visando gerar maior valor agregado do sistema para o cliente.

**Justificativa de Negócio**

**Custo benefício**

A elaboração de software voltando ao atendimento de negócios com fluxo de entrada de pedidos e saída de produto é um ramo com muita variedade de implantações e com poucos competidores fortes no mercado, como um pet-shop, pizzarias, restaurantes, etc. A utilização de um software de controle de entrada e saída traz ao comercio uma maior chance de m crescimento ascendente.

Com relação ao custo benefícios, a utilização de um software igual ao proposto é de suma importância já que como software é possível atender a um nicho maior de pessoas, amenizando a restrição restrições geográficas e de transporte. Com a criação de uma aplicação para uso mobile e web trata maior visibilidade ao negócio, trazendo benefícios como agilidade de recebimento de pedidos, divulgação do negócio, controle FIFO e outros mais.

**Ponto de equilíbrio**

Partindo do princípio que o software esta sendo implementado em novo negócio, a utilização do mesmo trará muitos benefícios de pronta entrega, um deles é a facilitação e maior controle do negócio, com isso, o break even é uma meta que será atingida rapidamente. Com maior controle destinado ao software fazer esses controles, seu negócio precisará de menos pessoas pensando em atender clientes, anotar pedidos e ter uma ralação frequente de atendimento ao consumidor.

Com o trabalho de receber pedidos destinado ao software, a quantidade de esforços pode ser maior alocada para recursos mais críticos, como fazer o que foi pedido pelo consumidor, trazendo assim maior velocidade de atendimento e finalização do pedido, trazendo feed-backs para o consumidor dos estados de se pedido, no caso de um petshop, quando seu cachorro começa a ser atendido, quando ele começa a tomar banho, e por fim quando ele está pronto.

**Causa/efeito**

Sabendo o principal problema do negócio, isso pode ser solucionado pela implementação de m software, delimitando e conhecendo muito bem o problema, soluciona-lo via software é uma questão de tempo, descobrindo a raiz do problema e solucionando-o etapa por etapa com o auxilio do software trará maior valor para seu negócio.

**Cadeia de valor**

A integração proporcionada pelo software pode ser desde uma pequena e simples parte do sistema, até uma integração completa, de ponto a ponto, integrando desde o fornecedor de matéria prima até o revendedor ou consumidor final. A integração pode proporcionar a inteligente compra de matéria prima, identificando o melhor valor de venda no mercado e também proporcionando o melhor custo de venda de seu produto.

Diminuindo custos fixos, isso trará maior competitividade de mercado, fazendo assim um negócio mais competitivo e com maior lucro.

**Oportunidade de investimento**

Como mencionado anteriormente, a oportunidade de investimento no ramo de software voltado à entrada e saída de pedidos é muito favorável, já que não existem grandes competidores para esse ramo. Entrando no mercado com um software bem consolidado e flexível as diversas aplicações possíveis, a chance de consolidação desse tipo de software é promissora.

**Pareto**

Sabe-se que cerca de 20% são as funcionalidades mais importantes e serão utilizadas durante toda a vida útil do software (importante ressaltar que cerca de 45% dessas funcionalidades nunca serão utilizadas). O ideal é separar quais são essas tarefas mais importantes e iniciar o desenvolvimento por elas (começando pela mais importante delas). Existem diversas vantagens em fazer isso:

* O software é entregue velozmente, partindo do princípio que reduzimos o que não é necessário, e estamos trabalhando apenas no que é estritamente necessário
* O cliente tem mais tempo hábil para validar se realmente o sistema resolve seu problema
* Aplicando um modelo de escopo aberto/negociável, existem chances muito altas de que o software seja entregue com alta qualidade funcional (faz o que realmente deve fazer)
* Partindo do mais importante, provavelmente o cliente perceberá que muitas das funcionalidades realmente não eram necessárias
* O cliente pode utilizar/rentabilizar o software com poucos dias ou semana de desenvolvimento
* O cliente ficará mais satisfeito com a eficiência e qualidade das entregas

**Payback**

O tempo de retorno desde o investimento inicial até aquele momento em que os rendimentos acumulados tornam-se iguais ao valor do investimento inicial. Este indicador fornece ao gerente a estimativa de quanto tempo vai levar (meses ou anos) até que ele recupere a aplicação/investimento inicial.

O PayBack é também visto como um indicador de risco de projeto por isso todo plano de projeto ou novo negócio deve ter como prioridade, minimizar seu PayBack e é importante ressaltar que o PayBack está relacionado a outros indicadores.

Para calcular o PayBack, é preciso entender o que é fluxo de caixa – pois será a partir do fluxo de caixa que você vai calcular o tempo para o investimento retornar. Fazer o fluxo de caixa significa que você vai controlar, equilibrar e destinar com qualidade tudo que entra e sai de valores da sua organização.

**Conclusão**

O RUP em muitos aspectos tem uma metodologia bem parecida com processos ágeis, mas ele se difere de processos ágeis em vários pontos. O RUP acabara forçando de certa forma um número grande de artefatos ao longo de seu processo e isso o diferencia de metodologias ágeis que costumam empregar a modelagem ágil com seu princípio de documentar apenas quando for necessário se comunicar mais especificamente.

As metodologias ágeis têm em suas iterações todas as etapas (Análise, Planejamento, Modelagem, Desenvolvimento e Teste), ou seja, as iterações são processos completos que resultarão em requisitos funcionando. No RUP, em cada iteração serão aplicadas disciplinas que corresponderão a essas etapas descritas acima, porém, o nível de aplicação depende do enfoque da fase que a iteração está ocorrendo. No caso do RUP, na fase de Elaboração, disciplinas relacionadas a modelagem aparecerão com maior aplicação nas iterações, mas essa mesma disciplina não tem grande aplicação nas iterações da fase de Transição.

Em questão ao escopo, nas metodologias ágeis, o escopo não é definido previamente ou por completo em uma fase qualquer. No RUP, há um grande foco na definição do escopo na primeira fase da metodologia, que é a fase de Concepção. Apesar do RUP pregar a realimentação de informações ao longo do processo para refinar os requisitos ao longo do processo, a fase de Concepção traz um foco maior a definição do prévio do escopo, que foge das características de uma metodologia ágil.

Não é difícil abordar o RUP como se fosse uma metodologia ágil, em vários pontos essas duas metodologias tem coisas em comum, mas em pontos como os citados acima, é possível perceber há diferença entre essas duas visões de como abordar um desenvolvimento de software, tendo que decidir pela qual usar, dependendo muito de qual tipo de software será construído e qual será seu tamanho.