

ENGENHARIA DE SOFTWARE

Professor Me. Vladimir Geraseev Junior

METODOLOGIAS ÁGEIS



METODOLOGIA X FRAMEWORK

- **METODOLOGIA:** FORNECE PRATICAMENTE TUDO QUE É NECESSÁRIO PARA CONDUÇÃO DE UM PROJETO. DIZ O QUE E COMO FAZER.
- **FRAMEWORK:** APENAS INDICA QUAL A TRAJETÓRIA, MAS NÃO INDICA EXATAMENTE COMO FAZER, EXISTINDO A POSSIBILIDADE DE SER EMPREGADO JUNTAMENTE COM OUTROS PROCESSOS E TÉCNICAS.

SURGIMENTO DAS METODOLOGIAS ÁGEIS

- NA DÉCADA DE 1990 COMEÇARAM A EMERGIR MODELOS DE DESENVOLVIMENTO;
- ALTERNATIVA A MÉTODOS DIRIGIDOS A PLANOS;
- PESQUISAS APONTAVAM QUE APENAS 16% DOS PROJETOS OBTINHAM SUCESSO;
- MODELOS TRADICIONAIS FOCO MAIOR NA ANÁLISE DE REQUISITOS;
- MODELOS ÁGEIS PRIORIZAM O SOFTWARE EM SI;

O QUE SÃO METODOLOGIAS ÁGEIS?

- UMA ALTERNATIVA À GESTÃO TRADICIONAL DE PROJETOS;
- BUSCAM PROMOVER UM PROCESSO DE GERENCIAMENTO DE PROJETOS QUE INCENTIVA A INSPEÇÃO E ADAPTAÇÃO FREQUENTE;
- INCENTIVA O MAIOR TRABALHO EM EQUIPE, A AUTO-ORGANIZAÇÃO, A COMUNICAÇÃO FREQUENTE, O FOCO NO CLIENTE E A ENTREGA DE VALOR;
- BASICAMENTE, OS **MÉTODOS ÁGEIS** SÃO UM CONJUNTO DE PRÁTICAS EFICAZES QUE SE DESTINAM A PERMITIR A ENTREGA RÁPIDA E DE ALTA QUALIDADE DO PRODUTO.





KANBAN - HISTÓRIA



KANBAN - HISTÓRIA

- NO FINAL DA DÉCADA DE 40, A TOYOTA COMEÇA E ESTUDAR SUPERMERCADOS COM A IDEIA DE APLICAR TÉCNICAS DE PRATELEIRA;
- OS ENGENHEIROS DA TOYOTA PERCEBERAM UM PROCESSO QUE REFLETIA DE ACORDO COM OS ANTERIORES;
- A PARTIR DE ENTÃO PASSARAM A USAR UM CARTÃO VISÃO(KANBAN) PARA SINALIZAR PASSOS EM SEU PROCESSO DE FABRICAÇÃO.











KANBAN - HISTÓRIA

- O KANBAN SÓ FOI UTILIZADO COMO UM MÉTODO DE DESENVOLVIMENTO ÁGIL EM 2007, NA CORBIS.
- O DIRETOR DE ENGENHARIA, DAVID ANDERSON E SUA EQUIPE PROJETAM O KANBAN PARA SUBSTITUIR A ABORDAGEM UTILIZADA ATÉ ENTÃO.
- COM O PASSAR DOS MESES FIZERAM UMA SERIE DE LANÇAMENTOS BEM SUCEDIDOS JUNTO COM MELHORIAS NO PROCESSO.
- DESDE ENTÃO O KANBAN VEM GANHANDO RESPEITO NA COMUNIDADE DE DESENVOLVIMENTO DE SOFTWARE E MAIS EMPRESAS PASSARAM A ADOTA-LO.

KANBAN - CONCEITO

- É UM MÉTODO PARA IMPLANTAÇÃO DE MUDANÇAS QUE NÃO DESCREVE PRÁTICAS ESPECÍFICAS.
- AO INVÉS, OFERECE PRINCÍPIOS QUE BUSCAM MELHORAR O DESEMPENHO E REDUZIR O DESPERDÍCIO.
- VISA MELHORAR O DESEMPENHO E REDUZIR O DESPERDÍCIO, ELIMINANDO ATIVIDADES QUE NÃO AGREGAM VALOR PARA A EQUIPE.
- O KANBAN AJUDA A ASSIMILAR E CONTROLAR O PROGRESSO DE SUAS TAREFAS DEVIDO A SUA NATUREZA VISUAL.

KANBAN - CONCEITO

Fazer	Fazendo	Feito
 	 	 
		 
		

KANBAN - CONCEITO

- FÁCIL DE SER IMPLEMENTADO;
- POUCO PRESCRITIVO;
- ADAPTÁVEL A QUALQUER AMBIENTE.

KANBAN - PRINCÍPIOS

VISUALIZAR O FLUXO DE TRABALHO (WORKFLOW)

- O MODELO VISUAL GERA BENEFÍCIOS COMO FOCO NO “TODO”, TRANSPARÊNCIA, IDENTIFICAÇÃO DE DESPERDÍCIOS;
- ENXERGANDO O CONTEXTO DO OUTRO SE AUMENTA A COMUNICAÇÃO E COLABORAÇÃO;
- O KANBAN PROPORCIONA UMA VISÃO AMPLA DO QUE ESTÁ SENDO FEITO, EM QUAL ETAPA SE ENCONTRA, QUANTO ESTÁ PRONTO E O QUANTO A EQUIPE CONSEGUE ENTREGAR.
- CONCEDE PREVISIBILIDADE, TENDO EM MÃOS MAIOR PLANEJAMENTO E CIÊNCIA DE QUANDO ASSUMIR RESPONSABILIDADES.

KANBAN - PRINCÍPIOS

LIMITAR A QUANTIDADE DE TRABALHO EM ANDAMENTO (WIP)

- AO LIMITAR O WIP, O RITMO DA EQUIPE SE TORNA EQUILIBRADO, SEM SE COMPROMETER COM MUITO TRABALHO DE UMA VEZ SÓ;
- REDUZ O TEMPO GASTO EM UM ITEM;
- EVITA PROBLEMAS COM ALTERNÂNCIA DE TAREFAS.

KANBAN - PRINCÍPIOS

GERENCIAR E MEDIR O FLUXO

- PODE-SE OTIMIZAR O KANBAN COLETANDO MÉTRICAS E INDICADORES DE PROBLEMAS FUTUROS;
- PARA SE DESCOBRIR ONDE É PRECISO OLHAR E ENTENDER COMO O TRABALHO ESTÁ FLUINDO, ANALISAR ÁREAS PROBLEMÁTICAS E IMPLEMENTANDO MUDANÇAS QUE FAVOREÇAM A MELHORIA.

KANBAN - PRINCÍPIOS

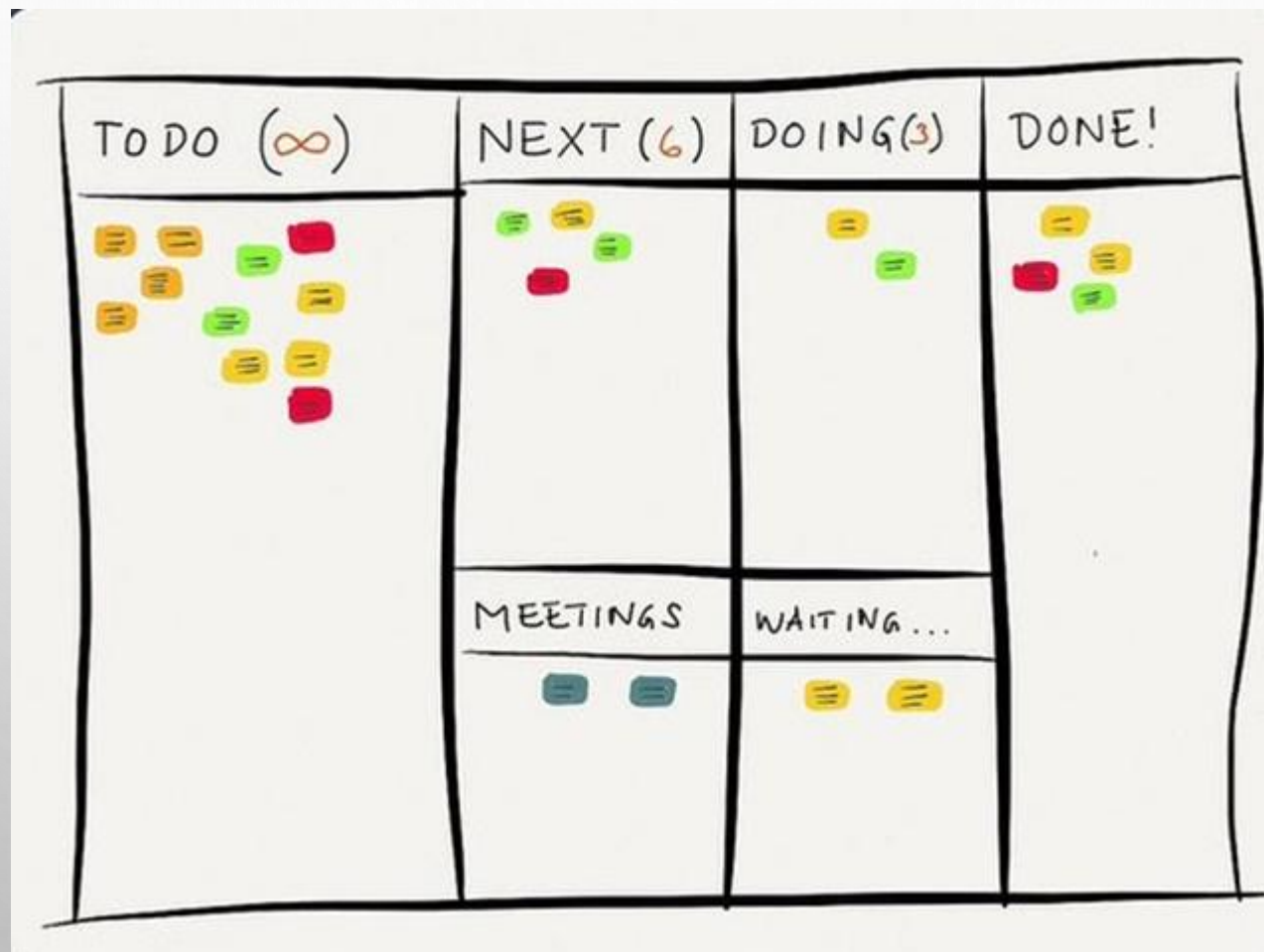
USAR MODELOS PARA RECONHECER OPORTUNIDADES DE MELHORIA

- ESTE PRINCÍPIO DIZ QUE SE VOCÊ NÃO ESTÁ MELHORANDO CONTINUAMENTE, MAS ESTÁ CUMPRINDO TODOS OS OUTROS REQUISITOS DO MÉTODO KANBAN, VOCÊ ESTÁ FAZENDO ERRADO;
- É COMO UTILIZAR UMA METODOLOGIA ÁGIL, MAS NÃO SER ÁGIL;
- O **KANBAN** SUGERE QUE MODELOS SEJAM USADOS PARA IMPLEMENTAR MUDANÇAS CONTÍNUAS, INCREMENTAIS E EVOLUTIVAS;
- EXEMPLO: TOC, SYSTEM THINKING, 3 MS.

KANBAN - IMPLEMENTAÇÃO

- BASICAMENTE O KANBAN POSSUI TRÊS CAMPOS:
 - TO DO (PARA FAZER);
 - DOING (EM EXECUÇÃO);
 - DONE (FINALIZADO).
- KANBAN PORT;
- OS CAMPOS SÃO ABASTECIDOS COM CARTÕES;
- CADA CARTÃO DEVE CONTER APENAS UM TAREFA.

KANBAN - IMPLEMENTAÇÃO



KANBAN - IMPLEMENTAÇÃO

- O KANBAN IMPLEMENTA CONCEITOS DA TEORIA DAS RESTRIÇÕES (TOC – *THEORY OF CONSTRAINTS*) ATRAVÉS DE UM MODELO DE SISTEMA “PUXADO”, PROCURANDO IDENTIFICAR E ADMINISTRAR RESTRIÇÕES QUE LIMITAM A PERFORMANCE DO SISTEMA;
- PRODUÇÃO EM UM SISTEMA DE *JUST-IN-TIME* (JIT).

O JUST-IN-TIME É UM SISTEMA DE ADMINISTRAÇÃO DA PRODUÇÃO QUE DETERMINA QUE NADA DEVE SER PRODUZIDO, TRANSPORTADO OU COMPRADO ANTES DA HORA EXATA. ELE PODE SER APLICADO EM QUALQUER ORGANIZAÇÃO, PARA REDUZIR ESTOQUES E OS CUSTOS DECORRENTES. ESSE SISTEMA TAMBÉM É UM DOS PILARES DO TOYOTISMO.

KANBAN - IMPLEMENTAÇÃO

- O KANBAN PODE SER UTILIZADO EM DIVERSAS ÁREAS, COMO POR EXEMPLO, DESENVOLVIMENTO DE SOFTWARE, GESTÃO DE TI, NOVOS NEGÓCIOS, DESIGN, FINANÇAS, MARKETING, OPERAÇÕES, ENTRE OUTRAS.

KANBAN - IMPLEMENTAÇÃO

- EQUIPE;
- IDENTIFICANDO ESTÁGIOS DE TRABALHO;
- PRIORIZAÇÃO;
- MEDIÇÃO E MELHORIA CONTÍNUA.

KANBAN - IMPLEMENTAÇÃO

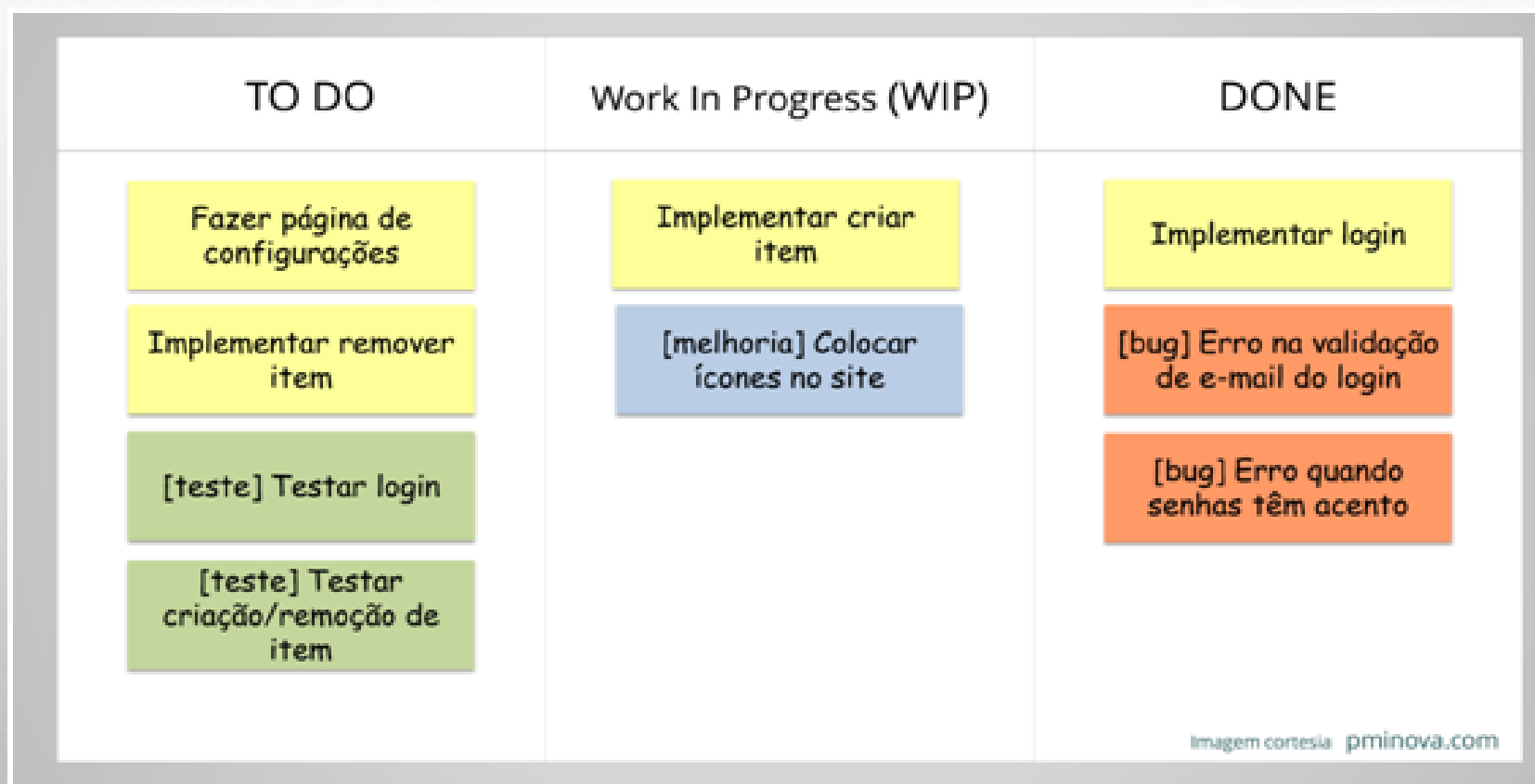
- **EQUIPE**

- É IMPORTANTE QUE A EQUIPE ESTEJA PREPARADA PARA ASSIMILAR OS CONCEITOS E PRINCÍPIOS DO KANBAN;
- APRESENTE OS CONCEITOS, PRINCÍPIOS, E UM EXEMPLO BÁSICO DE USO DO KANBAN E DOS BENEFÍCIOS QUE A EQUIPE PODERÁ OBTER COM SEU USO;
- SEMPRE É BOM LEMBRAR QUE, INDEPENDENTEMENTE DA FERRAMENTA, A EQUIPE É A PARTE MAIS IMPORTANTE. UMA EQUIPE CAPACITADA FUNCIONA BEM COM QUALQUER FERRAMENTA, ENQUANTO QUE UMA EQUIPE INCAPAZ GERALMENTE FALHA UTILIZANDO ATÉ MESMO O MELHOR DOS MÉTODOS.

KANBAN - IMPLEMENTAÇÃO

- **IDENTIFICANDO ESTÁGIOS DO TRABALHO**
 - IDENTIFIQUE OS ESTÁGIOS DE TRABALHO QUE SUA EQUIPE SEGUE PARA CONCLUIR UM PRODUTO, PROJETO OU SERVIÇO. O FLUXO MAIS COMUM COMEÇA EM “TO DO” E TERMINA EM “DONE”, GERALMENTE COM “WIP” (WORKING IN PROGRESS) NO MEIO, MAS PODE SER ALTERADO PARA O QUE MELHOR SE ADAPTAR AO SEU CONTEXTO;
 - IDENTIFIQUE AS CLASSES DE TRABALHO, COMO, POR EXEMPLO [USER STORIE], [BUG], [DEFEITO], [MELHORIA], [TESTE], [REQUISITO], ETC. QUE VÃO AJUDAR A CATEGORIZAR O TRABALHO E MELHOR ORGANIZAR O QUADRO;
 - TENDE DEFINIR LIMITES DE TEMPO QUE CARTÕES PODEM FICAR EM DETERMINADOS ESTÁGIOS, POR EXEMPLO, UM CARTÃO NÃO PODE FICAR MAIS QUE DUAS HORAS EM “TO DO” SE FOR DO TIPO [DEFEITO].

KANBAN - IMPLEMENTAÇÃO



Exemplo de kanban com tarefas

KANBAN - IMPLEMENTAÇÃO

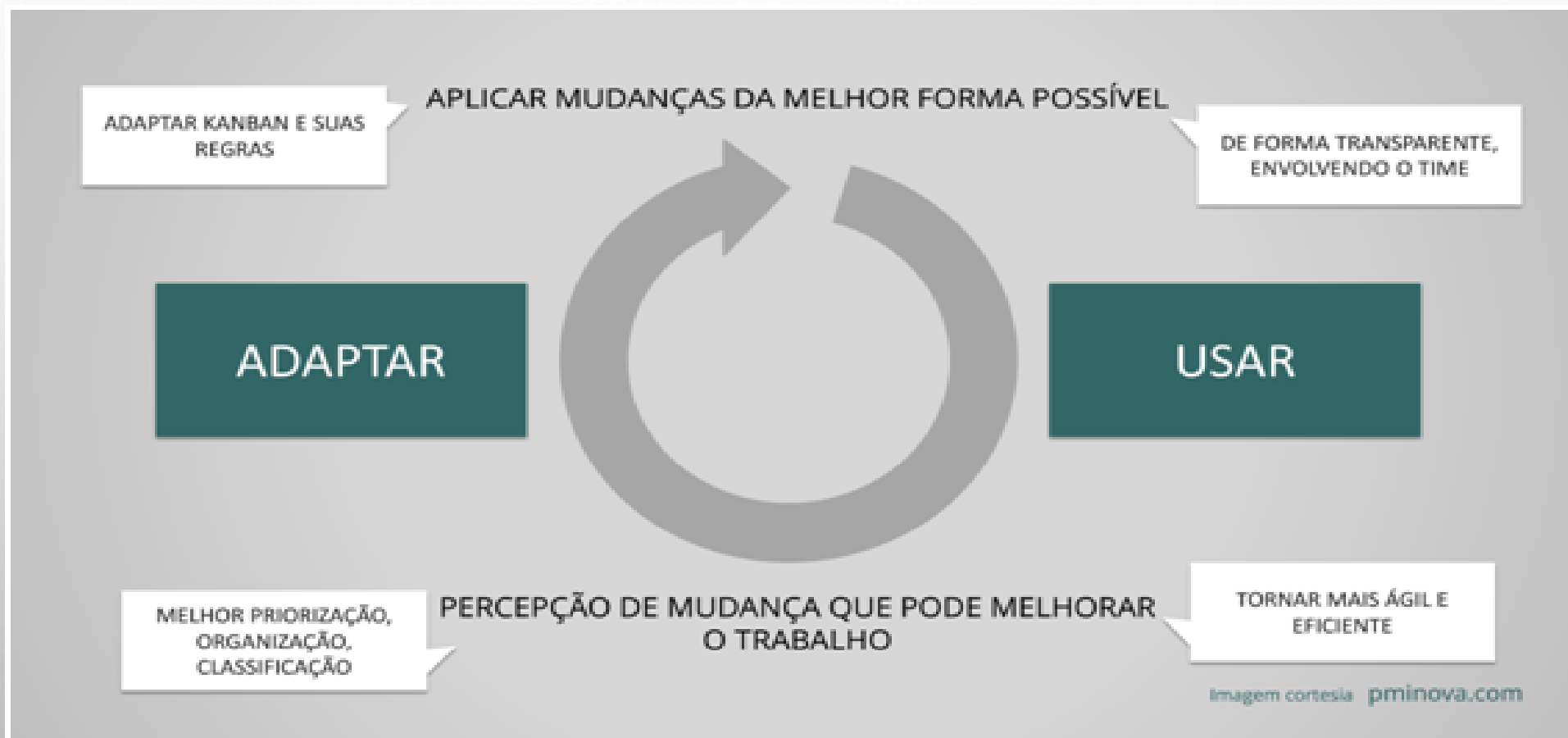
- **PRIORIZAÇÃO**

- POSICIONE O QUE É MAIS PRIORITÁRIO (TEM QUE SER FEITO ANTES) SEMPRE NA PARTE SUPERIOR DO KANBAN. SE DESEJAR, PODE SEPARAR O QUADRO EM CATEGORIAS, MAS MANTENDO ENTRE ELAS ESSA ESTRUTURA DE PRIORIZAÇÃO;
- MANTENHA UM CONTROLE CONSTANTE DO QUE É PRIORITÁRIO IDENTIFICANDO, SEMPRE QUE POSSÍVEL, MUDANÇAS NA ORDEM OU NOVOS CARTÕES QUE SEJAM NECESSÁRIOS. DESSA FORMA VOCÊ MELHORA A QUALIDADE E REDUZ OS CUSTOS, ELIMINANDO/ADIANDO O TRABALHO DESNECESSÁRIO.

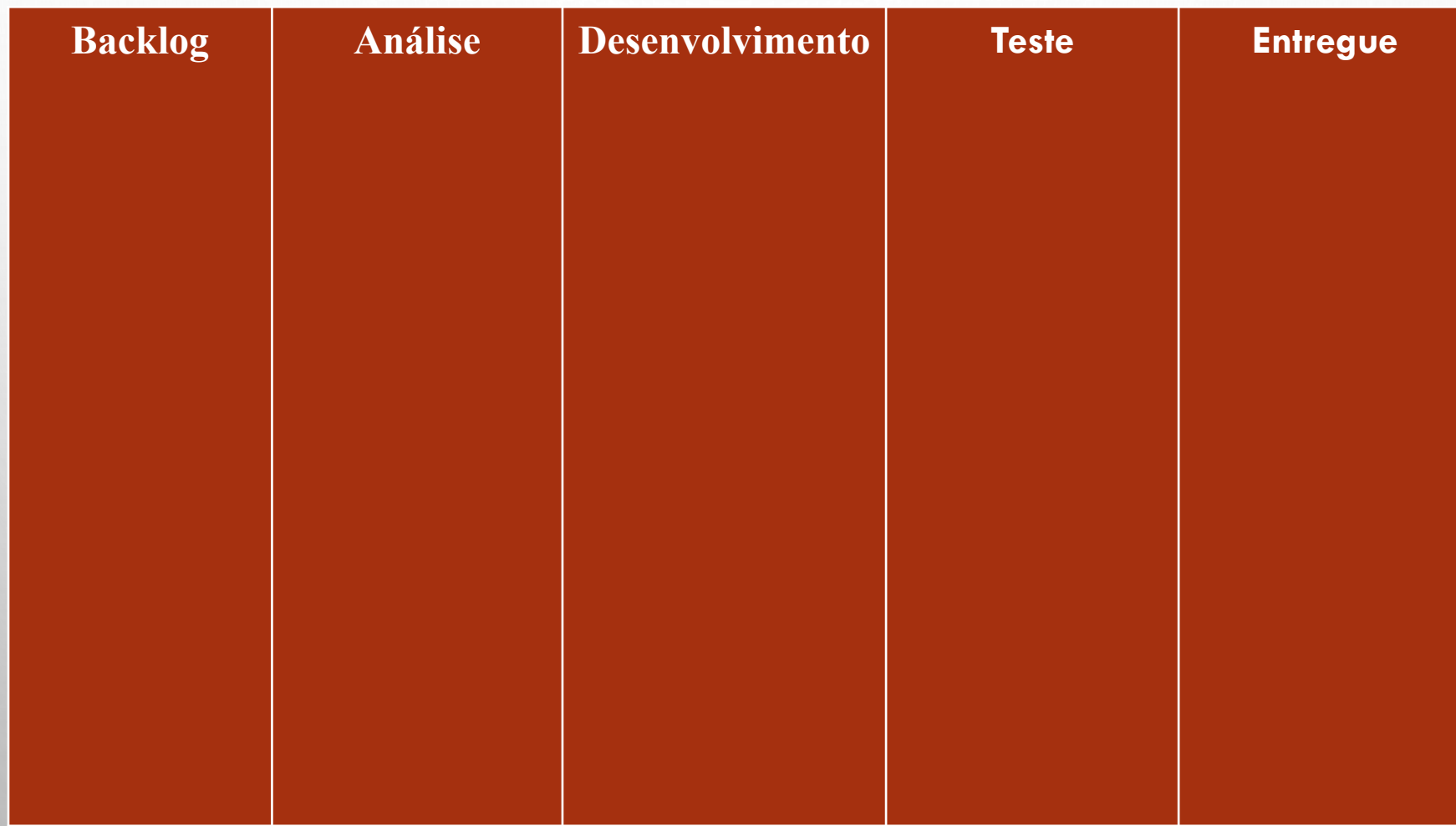
KANBAN - IMPLEMENTAÇÃO

- **MEDIÇÃO E MELHORIA CONTÍNUA**
 - ESTABELEÇA UM SISTEMA DE MEDIÇÃO SIMPLES COMO: TRABALHO QUE PRECISA SER FEITO E O QUE FOI FEITO ATÉ O INSTANTE;
 - SIMULE OS RISCOS DURANTE A EXECUÇÃO DO TRABALHO, PROCURANDO ENCONTRAR OS GARGALOS ANTES QUE ELES APAREÇAM. ISSO AJUDA A DETERMINAR PLANOS DE AÇÃO PREVENTIVOS, QUE DIMINUEM CONSIDERAVELMENTE OS RISCOS.

KANBAN - IMPLEMENTAÇÃO



KANBAN - IMPLEMENTAÇÃO



KANBAN - IMPLEMENTAÇÃO

FANESE - Desenvolvimento FANESE - Desenvolvimento ☆ Particular ... Mostrar M

To Do ...

Sistema de Processo Seletivo (Vestibular): Gerar relatórios de Aprovados, não matriculados, etc... GB LS

Controle Acadêmico: Criar um relatório de Quantidade de Alunos por Semestre com todos os status para os coordenadores e para Rosane. GB LS

Controle Acadêmico: Alterar o módulo de Reserva de Equipamento. Aguardar Neto informar todas as mudanças que ele precisa. 1 LS MS

Pós-Graduação: Dar a Opção de Alterar as informações de um Curso da Pós-Graduação enquanto ele ainda não foi solicitado para cadastro no RM. LS TR

Verificar o problema na Pesquisa no Ambiente da Biblioteca Virtual da Pearson.

Adicionar um cartão...

Priority ...

Informar ao usuário que ele precisa desativar o bloqueio de Popup referente ao Controle Acadêmico e para a Biblioteca Online. 1 LS TR

Desabilitar o Mural de recados para as Turmas a Distância. LS TR

Controle Acadêmico: Alterar as páginas que possuem iframe para validar a conexão com a internet. 1 GB LS

Controle Acadêmico Mobile: Android e IOS. 0/2 GB LS

Controle Acadêmico: Adicionar um módulo de manutenção para cadastro e carga de alunos. GB LS

Cursos de Extensão: Módulo novo (interno e externo) de Cursos de

Adicionar um cartão...

Doing ...

Controle Acadêmico: Alterar o layout do sistema. 1 GB LS

Integrar a senha do Controle Acadêmico com a senha de Rede da Fanese (Aluno e Professor). 4/7 GB LS

Revisar os relatórios do Controle Acadêmico, para alterar a forma de abertura de Popup para Nova Aba. 1 1 LS TR

Liberar o módulo de protocolo para o DTEC e Suporte. 1 0/1 LS TR

Módulo de Protocolo: Criação do módulo de Abertura de Protocolo pelo Aluno. 1 LS TR

Controle Acadêmico: Alteração no Módulo de Coordenador para

Adicionar um cartão...

Test ...

Verificar o tempo de sessão para o usuário do Controle Acadêmico no acesso fora da rede da FANESE. 3 LS

Controle Acadêmico: Novo Módulo de Avaliação. 1 LS MS

Adicionar um cartão...

Done ...

Controle Acadêmico: Alterar as telas de alteração de senha para mostrar os dados do curso do aluno e a foto do aluno e do professor. 3/3 GB LS

Controle Acadêmico: Verificar o problema enviado por e-mail por Patrícia referente ao fechamento de aula da disciplina TCC. GB LS

WebAula 2017/1: Dar carga nos alunos das Disciplinas a Distância na plataforma de EAD. 20 de fev LS

Adicionar um cartão...

Ativar o Windows
Acesse Configurações para ativar o Windows.



SCRUM - ORIGEM

- A ORIGEM DO TERMO SCRUM VEM DO RUGBY. É O MOMENTO DO JOGO QUANDO A EQUIPE ESTÁ TODA UNIDA COM UM ÚNICO PROPÓSITO EM UMA FORMAÇÃO ESPECÍFICA ONDE A PARTICIPAÇÃO DE TODOS É ESSENCIAL, A FALTA DE COMPROMETIMENTO DE UM MEMBRO PODE FAZER A FORMAÇÃO CAIR.
- A ANALOGIA DO TERMO DO RUGBY PARA O GERENCIAMENTO É SIMPLES. AMBOS SÃO FORMADOS POR UMA EQUIPE ESTRATEGICAMENTE DEFINIDA, TREINADA E ORGANIZADA COM UM OBJETIVO ÚNICO.



SCRUM - HISTORICO



SCRUM - OBJETIVO

- O SCRUM É UM *FRAMEWORK* DENTRO DO QUAL PESSOAS PODEM TRATAR E RESOLVER PROBLEMAS COMPLEXOS E ADAPTATIVOS, ENQUANTO PRODUTIVA E CRIATIVAMENTE ENTREGAM PRODUTOS COM O MAIS ALTO VALOR POSSÍVEL.

SCRUM - CARCTERISTICAS

- BASEADO NO EMPIRISMO;
- TRABALHA DE FORMA ITERATIVA E INCREMENTAL ;
- EQUIPES MULTIFUNCIONAIS E AUTO ORGANIZÁVEIS;

FOCO EM:

- PRIORIDADE: SABER ONDE COMEÇAR E O QUE É MAIS PRIORITÁRIO;
- OBJETIVIDADE: METAS MENORES ATINGÍVEIS E CLARAS;
- VISIBILIDADE: DO QUE ESTÁ COMPLETO E AS PENDÊNCIAS;

SCRUM – HISTORIA DE USUÁRIOS

- **UMA HISTÓRIA DE USUÁRIO SCRUM É UMA DESCRIÇÃO CONCISA E CENTRADA NO USUÁRIO DE UMA FUNCIONALIDADE OU RECURSO DESEJADO EM UM PROJETO ÁGIL SCRUM, SERVINDO COMO UMA UNIDADE FUNDAMENTAL DE TRABALHO NA ESTRUTURA SCRUM. NORMALMENTE ESCRITA DA PERSPECTIVA DO USUÁRIO FINAL, UMA HISTÓRIA DE USUÁRIO ARTICULA O QUEM, O QUÊ E O PORQUÊ DE UM REQUISITO ESPECÍFICO, ENFATIZANDO A COLABORAÇÃO E A COMUNICAÇÃO ENTRE EQUIPES MULTIFUNCIONAIS.**

SCRUM – HISTORIA DE USUÁRIOS

UMA HISTÓRIA DE USUÁRIO SEGUE UM MODELO SIMPLES QUE INCLUI UM PAPEL DO USUÁRIO (ATOR), UMA AÇÃO OU OBJETIVO DESEJADO E UM BENEFÍCIO OU VALOR.

SCRUM – HISTORIA DE USUÁRIOS

UMA HISTÓRIA DE USUÁRIO SEGUE UM MODELO SIMPLES QUE INCLUI UM PAPEL DO USUÁRIO (ATOR), UMA AÇÃO OU OBJETIVO DESEJADO E UM BENEFÍCIO OU VALOR.

SCRUM – HISTORIA DE USUÁRIOS

UMA HISTÓRIA DE USUÁRIO SEGUE UM MODELO SIMPLES QUE INCLUI UM PAPEL DO USUÁRIO (ATOR), UMA AÇÃO OU OBJETIVO DESEJADO E UM BENEFÍCIO OU VALOR.

EXEMPLO:

COMO ATOR, EU QUERO/DEVO REALIZAR UM OBJETIVO.

SCRUM – HISTORIA DE USUÁRIOS

- **ATOR:** O "PROPRIETÁRIO" DA HISTÓRIA DE USUÁRIO. GERALMENTE É UM USUÁRIO, MAS RECOMENDA-SE SER MAIS ESPECÍFICO. AO USAR ATORES ESPECÍFICOS (POR EXEMPLO, "ADMINISTRADOR", "CLIENTE CONECTADO", "VISITANTE NÃO AUTENTICADO"), FICA MAIS FÁCIL ENTENDER E CONTEXTUALIZAR A HISTÓRIA DE USUÁRIO.
- **AÇÃO:** O QUE O ATOR DESEJA FAZER. SE FOR UMA AÇÃO OBRIGATÓRIA, PODE SER PRECEDIDA POR "DEVE". CASO CONTRÁRIO, USA-SE "DESEJA".
- **REALIZAÇÃO:** O QUE O ATOR DESEJA ALCANÇAR AO REALIZAR A AÇÃO. É O RESULTADO DA EXECUÇÃO DA AÇÃO, VISTO DO PONTO DE VISTA DO ATOR.

SCRUM – HISTORIA DE USUÁRIOS

EXEMPLOS

- Como leitor ávido, desejo receber recomendações de livros personalizadas com base no meu histórico de leitura, para que eu possa descobrir novos livros que estejam alinhados com meus interesses.
- Como proprietário de uma casa inteligente, quero controlar todos os meus dispositivos inteligentes a partir de um único aplicativo, para poder gerenciar facilmente a tecnologia da minha casa.
- Como proprietário de um restaurante, quero atualizar os itens do meu cardápio em tempo real no meu site, para que meus clientes tenham informações precisas.

SCRUM - PAPEIS

- **PRODUCT OWNER**
- **SCRUM MASTER**
- **TIME DE DESENVOLVIMENTO**

PRODUCT OWNER - GERENCIA O PRODUCT BACKLOG

RESPONSÁVEL POR MAXIMIZAR O VALOR DO PRODUTO E DO TRABALHO DO TIME.

- REPRESENTA TODOS OS DEMAIS STAKEHOLDERS;
- RESPONSÁVEL POR DEFINIR A FUNCIONALIDADE DO PRODUTO;
- RESPONSÁVEL PELO GERENCIAMENTO DO PRODUCT BACKLOG;
- RESPONSÁVEL PELO ACEITE DO PRODUTO;
- NÃO É UM COMITÊ MAS UMA ÚNICA PESSOA;
- É O ÚNICO RESPONSÁVEL PELA MANUTENÇÃO DO BACKLOG;
- NÃO PODE SER O SCRUM MASTER;
- DEFINE PRA ONDE O TIME DEVE IR MAS NÃO COMO CHEGAR LÁ.

SCRUM MASTER- GERENCIA O PROCESSO

RESPONSÁVEL POR GARANTIR QUE O SCRUM SEJA ENTENDIDO E APLICADO.

- GARANTIR QUE AS PRÁTICAS DO SCRUM ESTÃO SENDO SEGUIDAS;
- TREINAR A EQUIPE PARA QUE SEJA AUTO GERENCIÁVEL E MULTIFUNCIONAL;
- GARANTIR QUE O PRODUCT OWNER ESTÁ DESEMPENHANDO SEU TRABALHO;
- REMOVER IMPEDIMENTOS, VISÍVEIS E NÃO-VISÍVEIS, INTERNOS E EXTERNOS;
- MOTIVAR E MANTER O TRABALHO EM EQUIPE;
- ASSEGURAR A MELHORIA CONTÍNUA;
- GERENCIAR AS EXPECTATIVAS.

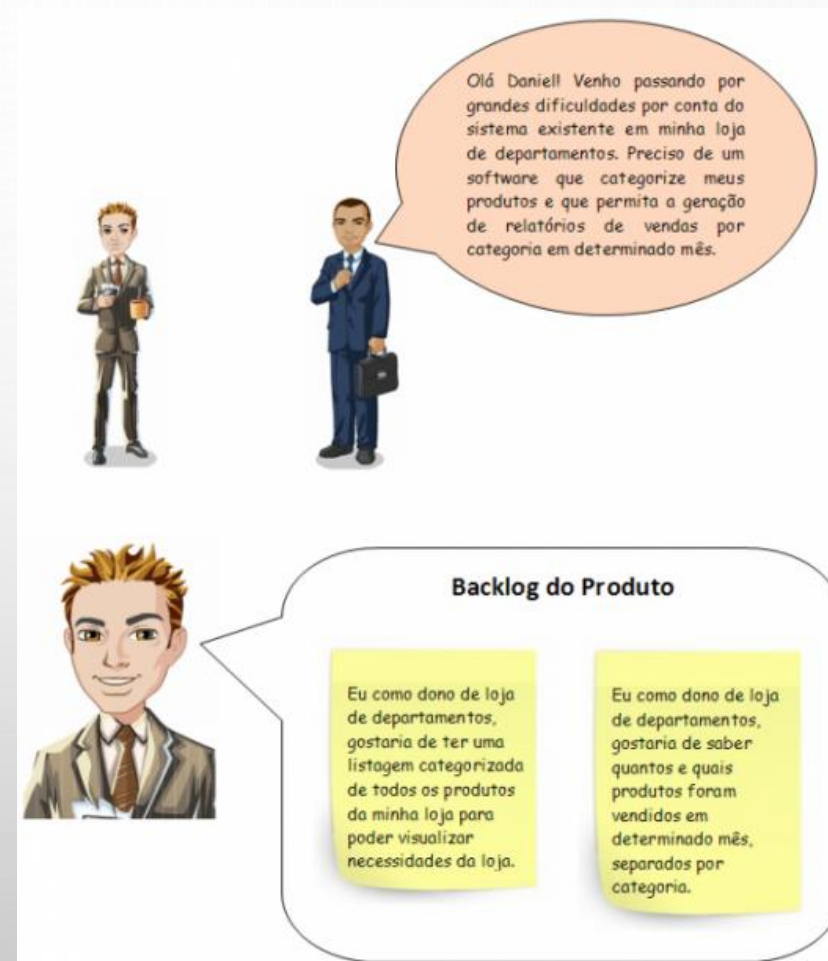
TIME DE DESENVOLVIMENTO - GERENCIAM A SI MESMO

REALIZA O TRABALHO DE ENTREGAR UMA VERSÃO USÁVEL QUE POTENCIALMENTE INCREMENTA O PRODUTO.

- RESPONSÁVEL POR TRANSFORMAR OS REQUISITOS EM UM PRODUTO PARA O USUÁRIO;
- RESPONSÁVEL PELA ENTREGA DO PRODUTO E PELA QUALIDADE DO MESMO;
- COMPARTILHAM AS RESPONSABILIDADES;
- TRABALHAM EM EQUIPE PARA DESENHAR, CODIFICAR, TESTAR E O PRODUTO DE SW;
- COMPOSTO POR MEMBROS NECESSÁRIOS PARA TOMAR DECISÕES E REALIZAR AS TAREFAS;
- EM MÉDIA DE 6 A 10 PESSOAS.

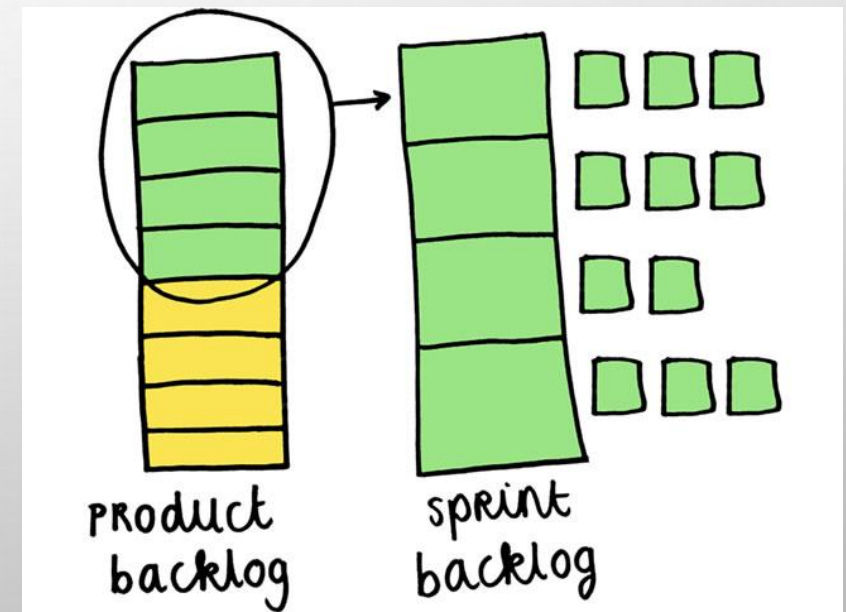
BACKLOG DO PRODUTO

- É UMA LISTA ORDENADA DE TUDO QUE DEVE SER NECESSÁRIO NO PRODUTO;
- É UMA ORIGEM ÚNICA DOS REQUISITOS PARA QUALQUER MUDANÇA A SER FEITA NO PRODUTO;
- NUNCA ESTÁ COMPLETO – É DINÂMICO;



BACKLOG DO SPRINT

- É UM CONJUNTO DE ITENS DE BACKLOG DO PRODUTO SELECIONADOS PARA A SPRINT, E O PLANO PARA ATINGIR O OBJETIVO DA SPRINT;
- PODE SER ALTERADO DURANTE TODA A SPRINT;



EVENTOS SCRUM

- SÃO PREVIAMENTE DEFINIDOS;
- SÃO UTILIZADOS PARA CRIAR UMA ROTINA E DIMINUIR A NECESSIDADE DE REUNIÕES NÃO DEFINIDAS NO SCRUM;
- TODO EVENTO TEM UMA DURAÇÃO MÁXIMA;
- SPRINT
 - REUNIÃO DE PLANEJAMENTO DA SPRINT;
 - REUNIÃO DIÁRIA;
 - TRABALHO DE DESENVOLVIMENTO;
 - REVISÃO DA SPRINT;
 - RETROSPECTIVA DA SPRINT;

EVENTOS SCRUM

- **SPRINT:**
 - PLANO DE PROJETO (FLEXÍVEL) É DEFINIDO;
 - DURAÇÃO: UM MÊS OU MENOS;

- **CANCELAMENTO DA SPRINT:**
 - PODE OCORRER SE O SEU OBJETIVO SE TORNAR OBSOLETO (PRODUCT OWNER);
 - CONSOME RECURSOS, SÃO TRAUMÁTICOS E INCOMUNS.

EVENTOS SCRUM

- REUNIÃO DE PLANEJAMENTO DA SPRINT:
 - CRIAÇÃO DE UM PLANO COM A COLABORAÇÃO DE TODO O TIME SCRUM;
 - DURAÇÃO: OITO HORAS – PARA SPRINTS DE UM MÊS;
 - SCRUM MASTER;

- QUESTIONAMENTOS:
 - O QUE PODE SER ENTREGUE COMO RESULTADO DO INCREMENTO DA PRÓXIMA SPRINT? (TIME DE DESENVOLVIMENTO)
 - COMO O TRABALHO NECESSÁRIO PARA ENTREGAR O INCREMENTO SERÁ REALIZADO? (TIME DE DESENVOLVIMENTO)

EVENTOS SCRUM

- REUNIÃO DIÁRIA: TIME DE DESENVOLVIMENTO + SCRUM MASTER
 - DURAÇÃO: 15 MINUTOS;
 - OBJETIVO: SINCRONIZAR AS ATIVIDADES E CRIAR O PLANO PARA AS PRÓXIMAS 24HORAS;
 - REALIZAÇÃO SEMPRE NO MESMO LOCAL E HORÁRIO;

ESCLARECIMENTOS:

- O QUE EU FIZ ONTEM QUE AJUDOU O TIME DE DESENVOLVIMENTO A ATENDER A META DA SPRINT?
- O QUE EU FAREI HOJE PARA AJUDAR O TIME DE DESENVOLVIMENTO A ATENDER A META DA SPRINT?
- EU VEIO ALGUM OBSTÁCULO QUE IMPEÇA A MIM OU O TIME DE

EVENTOS SCRUM

- REVISÃO DA SPRINT:
 - DURAÇÃO: 4 HORAS – SPRINT DE UM MÊS;
 - OBJETIVO: VERIFICAR O INCREMENTO, E ADAPTAR O BACKLOG DO PRODUTO (SE NECESSÁRIO);
 - RESULTADO: SPRINT + BACKLOG DO PRODUTO REVISADOS;

EVENTOS SCRUM

- RETROSPECTIVA DA SPRINT:
 - DURAÇÃO: 3 HORAS – SPRINT DE UM MÊS;
 - OBJETIVO:
- INSPECIONAR COMO A ÚLTIMA SPRINT FOI EM RELAÇÃO ÀS PESSOAS, AO RELACIONAMENTOS, AOS PROCESSOS E FERRAMENTAS;
- IDENTIFICAR E ORDENAR OS PRINCIPAIS ITENS QUE FORAM BEM E AS POTENCIAIS MELHORIAS;
- CRIAR UM PLANO PARA IMPLEMENTAR MELHORIAS NO MODO QUE O SCRUM FAZ SEU TRABALHO;
 - AO FINAL: MELHORIAS DEVEM SER LEVANTADAS PARA SEREM APLICADAS NAS PRÓXIMAS SPRINTS.

ARTEFATOS DO SCRUM

- INCREMENTO:
 - É A SOMA DE TODOS OS ITENS DO BACKLOG DO PRODUTO COMPLETADOS DURANTE A SPRINT;
 - AO FINAL DE UMA SPRINT, UM NOVO INCREMENTO DEVE ESTÁ “PRONTO”;
 - “PRONTO”: VERSÃO INCREMENTAL POTENCIALMENTE UTILIZÁVEL.



HISTÓRIA

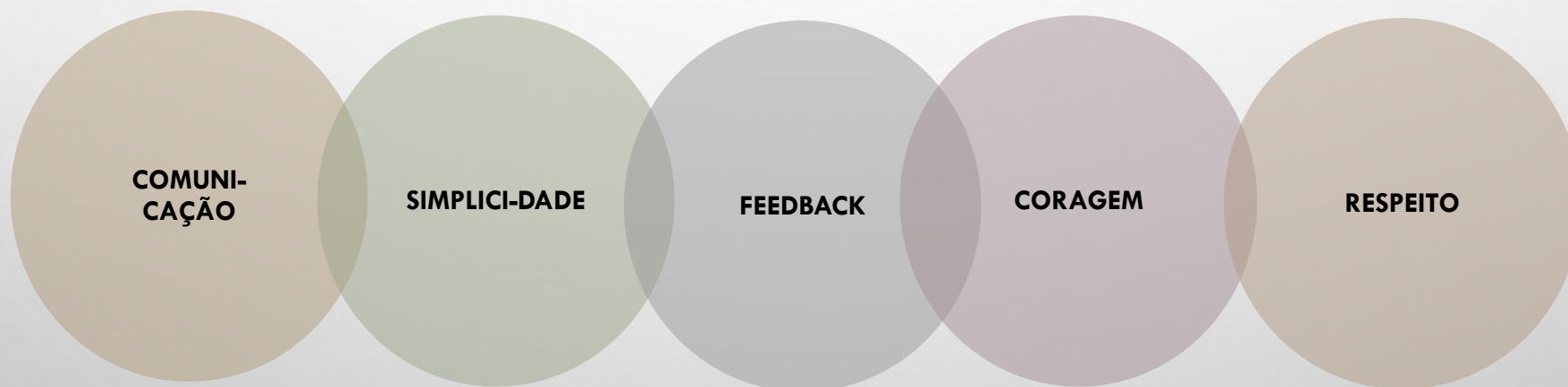
- TEM MARCO DE CRIAÇÃO EM 1996 POR KENT BECK E WARD CUNNINGHAM, CONSIDERADA UMA METODOLOGIA NOVA E QUE ESTÁ EM EVOLUÇÃO;
- PORÉM A JUNÇÃO DE PRINCÍPIOS E BOAS PRÁTICAS DE PROGRAMAÇÃO SÃO FRUTOS DE UM PROCESSO DE EVOLUÇÃO DE PELO MENOS UMA DÉCADA EM QUE KENT E WARD TRABALHARAM NA TEKTRONIXS, INC. COMO CONSULTORES DE PROBLEMAS EM SMALLTALK.

CONCEITO GERAL.

- FOCO EM AGILIDADE DE EQUIPES E QUALIDADE DE PROJETOS, APOIADA EM VALORES:
 - CORAGEM;
 - SIMPLICIDADE;
 - COMUNICAÇÃO;
 - FEEDBACK.

CONCEITO

- SE SUSTENTA EM CINCO VALORES, QUE SÃO:



- E reflete em cinco princípios, que são:



CORAGEM

ELA É NECESSÁRIA PARA QUE REALMENTE SE APLIQUE XP COMO DEVE SER APLICADO. EXEMPLOS DE ATITUDE QUE EXIGEM CORAGEM SÃO:

- ALTERAR CÓDIGO JÁ ESCRITO E QUE ESTÁ FUNCIONANDO;
- JOGAR CÓDIGO FORA E REESCREVER TUDO DE NOVO;
- PERMITIR CÓDIGO COMPARTILHADO POR TODOS.

ESTES EXEMPLOS DE ATITUDES PODEM SER NECESSÁRIOS PARA TRAZER MELHORIAS AO PROJETO E NÃO DEVEM SER EVITADAS SIMPLEMENTE DEVIDO AO MEDO DE TENTÁ-LAS.

SIMPLICIDADE

- XP SUGERE QUE CADA MEMBRO DA EQUIPE ADOTE A SOLUÇÃO MAIS FÁCIL QUE POSSA FUNCIONAR. O OBJETIVO É FAZER AQUILO QUE É MAIS SIMPLES HOJE E CRIAR UM AMBIENTE EM QUE O CUSTO DE MUDANÇAS NO FUTURO SEJA BAIXO.
- O OBJETIVO DESSA ABORDAGEM ADOTADA POR XP É EVITAR A CONSTRUÇÃO ANTECIPADA DE FUNCIONALIDADES, QUE ACABAM MUITAS VEZES NEM SENDO USADAS.

COMUNICAÇÃO

- XP FOCA EM CONSTRUIR UM ENTENDIMENTO PESSOA-A-PESSOA DO PROBLEMA, COM O USO MÍNIMO DE DOCUMENTAÇÃO FORMAL E COM O USO MÁXIMO DE INTERAÇÃO “CARA-A-CARA” ENTRE AS PESSOAS ENVOLVIDAS NO PROJETO.



FEEDBACK

- OS PROGRAMADORES OBTÊM FEEDBACK SOBRE A LÓGICA DOS PROGRAMAS ESCRIVENDO E EXECUTANDO CASOS DE TESTE. OS CLIENTES OBTÊM FEEDBACK ATRAVÉS DOS TESTES FUNCIONAIS CRIADOS PARA TODAS AS HISTÓRIAS (CASOS DE USO SIMPLIFICADOS).
- O FEEDBACK É IMPORTANTE, POIS POSSIBILITA QUE AS PESSOAS APRENDAM CADA VEZ MAIS SOBRE O SISTEMA E ASSIM CORRIJAM OS ERROS E MELHOREM O SISTEMA.

PRÁTICAS DA XP

- EXTREME PROGRAMMING É DINÂMICO E FLEXÍVEL, PORÉM É NECESSÁRIO MUITA DISCIPLINA PARA USÁ-LA EM UM PROJETO. PARA DEMONSTRAR ISSO, TEMOS UM CONJUNTO SUGERIDO DE "BOAS PRÁTICAS" EM PROJETOS USANDO XP.



O JOGO DO PLANEJAMENTO (PLANNING GAME)

O PLANEJAMENTO DE UM RELEASE E DAS ALTERAÇÕES SÃO FEITOS COM BASE NAS HISTÓRIAS E CONTA COM A COLABORAÇÃO DE TODA A EQUIPE DE DESENVOLVIMENTO, INCLUSIVE O CLIENTE, DIVIDIDOS EM DOIS PAPÉIS:

- **NEGÓCIO:** PARTICIPAM AS PESSOAS QUE MAIS ENTENDEM SOBRE O NEGÓCIO E QUE POSSAM ESTABELECEER PRIORIDADES PARA AS FUNCIONALIDADES A SEREM ENTREGUES.
- **TÉCNICO:** PARTICIPAM AS PESSOAS QUE IRÃO IMPLEMENTAR AS FUNCIONALIDADES DESCRITAS. OS TÉCNICOS ESTIMAM QUAL O ESFORÇO E RISCOS ENVOLVIDOS PARA IMPLEMENTAR AS FUNCIONALIDADES E COMUNICAM AO PESSOAL DE NEGÓCIOS.

PEQUENAS VERSÕES (SMALL RELEASES)

- CONFORME AS INTERAÇÕES SÃO CONCLUÍDAS, O CLIENTE RECEBE PEQUENAS VERSÕES (RELEASES) DO SISTEMA, VISANDO COM QUE SEJA COLOCADO EM PRÁTICA E VALIDADO AQUILO QUE ESTÁ SENDO IMPLEMENTADO;
- ISTO TAMBÉM PERMITE QUE MAIS CEDO POSSAM SER DETECTADAS NECESSIDADES DE ALTERAÇÕES DE REQUISITOS NO SOFTWARE.

METÁFORA (METAPHOR)

- A INTENÇÃO DA METÁFORA É OFERECER UMA VISÃO GERAL DO SISTEMA, EM UM FORMATO SIMPLES, QUE POSSA SER COMPARTILHADA POR CLIENTES E PROGRAMADORES;
- A IDEIA DA METÁFORA É QUE SEJA FEITA UMA ANALOGIA ENTRE O SISTEMA QUE ESTÁ SENDO DESENVOLVIDO E UM SISTEMA, NÃO NECESSARIAMENTE DE SOFTWARE, QUE TODOS ENTENDAM, COM O OBJETIVO DE OBTER UM “VOCABULÁRIO COMUM” PARA A POSTERIOR CRIAÇÃO DE NOMES DE CLASSES, SUBSISTEMAS, MÉTODOS, ENTRE OUTROS.
 - **EXEMPLO:** "VAMOS CHAMAR NOSSO PROJETO DE 'CARTÃO DE PONTO', PARA UM SISTEMA QUE GERENCIE AS BATIDAS DE PONTO DE FUNCIONÁRIOS, GERANDO O PROVISIONAMENTO FINANCEIRO E MENSAL PARA MÓDULO DE FOLHA DE PAGAMENTO".

PROJETO SIMPLES (SIMPLE DESIGN)

ESTA PRÁTICA TEM O INTUITO DE ENFATIZAR QUE O PROJETO SIMPLES DEVE SE CONCENTRAR EM SOLUÇÕES SIMPLES E BEM ESTRUTURADAS PARA OS PROBLEMAS DE HOJE E QUE NÃO SE DEVE PERDER TEMPO INVESTINDO EM SOLUÇÕES GENÉRICAS QUE PROCUREM ATENDER A FUNCIONALIDADES FUTURAS, POIS COMO OS REQUISITOS MUDAM CONSTANTEMENTE TAIS SOLUÇÕES GENÉRICAS PODEM NÃO SER MAIS A REALIDADE DO FUTURO.

ALGUMAS CARACTERÍSTICAS DE PROJETO SIMPLES:

- TODOS OS TESTES EXECUTAM COM SUCESSO;
- O PROJETO EXPRESSA A IDEIA QUE O PROGRAMADOR TEVE;
- O PROJETO NÃO POSSUI LÓGICA DUPLICADA;
- O PROJETO CONTÉM O MENOR NÚMERO POSSÍVEL DE CLASSES E MÉTODOS.

TESTE CONSTANTES (TEST FIRST DESIGN)

- OS TESTES EM XP SÃO FEITOS ANTES DA PROGRAMAÇÃO. EXISTEM DOIS TIPOS DE TESTE: TESTE DE UNIDADE E TESTE FUNCIONAL.
 - **TESTES DE UNIDADE:** SÃO FEITOS PARA VERIFICAR TUDO QUE POSSA DAR ERRADO. NÃO É PRECISO ESCREVER TESTES DE UNIDADE PARA TODOS OS MÉTODOS. OS TESTES UNITÁRIOS SÃO AUTOMATIZADOS, E TODA VEZ QUE O PROGRAMADOR ESCREVER CÓDIGO, ELE IRÁ VERIFICAR SE O SISTEMA PASSA EM TODOS OS TESTES
 - **TESTES FUNCIONAIS:** SÃO USADOS PARA VERIFICAÇÃO, JUNTO AO CLIENTE, DO SISTEMA COMO UM TODO. OS TESTES SERVEM COMO UM MECANISMO PARA ASSEGURAR QUE O SISTEMA ESTÁ SEMPRE RODANDO LIVRE DE ERROS, E TAMBÉM SERVEM PARA DAR FEEDBACK AOS PROGRAMADORES E CLIENTES SOBRE AS FALHAS ENCONTRADAS.

REFATORAÇÃO (REFACTORING)

- SÃO CONSTANTES MELHORIAS NO PROJETO DO SOFTWARE PARA AUMENTAR SUA CAPACIDADE DE SE ADAPTAR A MUDANÇAS. A REFATORAÇÃO CONSISTE EM APLICAR UMA SÉRIE DE PASSOS PARA MELHORAR O PROJETO DO CÓDIGO EXISTENTE, TORNANDO-O MAIS SIMPLES E MELHOR ESTRUTURADO, SEM ALTERAR SUA FUNCIONALIDADE.
- A CADA NOVA FUNCIONALIDADE ADICIONADA, É TRABALHADO O DESIGN DO CÓDIGO ATÉ FICAR NA SUA FORMA MAIS SIMPLES, MESMO QUE ISSO IMPLIQUE EM "MEXER" EM UM CÓDIGO QUE ESTEJA EM FUNCIONAMENTO.

PROGRAMAÇÃO EM PARES (PAIR PROGRAMMING)

- TODO O CÓDIGO PRODUZIDO EM XP É ESCRITO POR UM PAR DE PROGRAMADORES, QUE POSSUEM PAPÉIS DISTINTOS, SENTADOS LADO-A-LADO E OLHANDO PARA O COMPUTADOR.
- UM PARCEIRO SERÁ RESPONSÁVEL PELA CODIFICAÇÃO E PENSARÁ NOS ALGORITMOS E NA LÓGICA DE PROGRAMAÇÃO. O OUTRO PARCEIRO OBSERVA O CÓDIGO PRODUZIDO E TENTA PENSAR MAIS ESTRATEGICAMENTE EM COMO MELHORÁ-LO E TORNÁ-LO MAIS SIMPLES, ALÉM DE APONTAR POSSÍVEIS ERROS E PONTOS DE FALHA.
- ALÉM DISSO, AS DUPLAS SÃO CONSTANTEMENTE TROCADAS E OS PAPÉIS TAMBÉM, COM O OBJETIVO DE QUE TODOS OS MEMBROS DA EQUIPE POSSAM TER CONHECIMENTO SOBRE TODAS AS PARTES DO SISTEMA.

PROPRIEDADE COLETIVA DO CÓDIGO (COLLECTIVE CODE OWNERSHIP)

- A PROGRAMAÇÃO EM PARES ENCORAJA DUAS PESSOAS A TRABALHAREM JUNTAS PROCURANDO ATINGIR O MELHOR RESULTADO POSSÍVEL.
- A PROPRIEDADE COLETIVA ENCORAJA A EQUIPE INTEIRA A TRABALHAR MAIS UNIDA EM BUSCA DE QUALIDADE NO CÓDIGO FAZENDO MELHORIAS E REFATORAÇÃO EM QUALQUER PARTE DO CÓDIGO A QUALQUER TEMPO.
- A PRINCÍPIO PODE-SE PENSAR QUE ESTA PRÁTICA POSSA GERAR PROBLEMAS, MAS COMO TODOS DEVEM RESPEITAR UM PADRÃO DE CODIFICAÇÃO E DEVEM REALIZAR TODOS OS TESTES PARA VERIFICAÇÃO DE ERROS, ESTA ATIVIDADE É FEITA DE UMA FORMA CONTROLADA E PODE SER FACILITADA COM O USO DE UMA FERRAMENTA DE CONTROLE DE VERSÃO.

PADRÕES DE CODIFICAÇÃO (CODING STANDARDS)

- COMO A XP PREGA A PROPRIEDADE COLETIVA DE CÓDIGO, ONDE TODOS PODEM ALTERAR E FAZER REFATORAÇÃO DE QUALQUER PARTE DO CÓDIGO A QUALQUER MOMENTO, ENTÃO É MAIS DO QUE NECESSÁRIO QUE SE TENHA PADRÕES DE CODIFICAÇÃO. O OBJETIVO É QUE TODOS PROGRAMEM DA MESMA FORMA, FACILITANDO O ENTENDIMENTO DO CÓDIGO E AS ALTERAÇÕES.

INTEGRAÇÃO CONTÍNUA (CONTINUOUS INTEGRATION)

- O CÓDIGO DAS FUNCIONALIDADES IMPLEMENTADAS PODE SER INTEGRADO VÁRIAS VEZES AO DIA. UM MODO SIMPLES DE FAZER ISSO É TER UMA MÁQUINA DEDICADA (SERVIDOR) PARA INTEGRAÇÃO. QUANDO A MÁQUINA ESTIVER LIVRE, UM PAR COM CÓDIGO A INTEGRAR OCUPA A MÁQUINA, CARREGA A VERSÃO CORRENTE DO SISTEMA, CARREGA AS ALTERAÇÕES QUE FIZERAM (RESOLVENDO OS ERROS), E RODAM OS TESTES ATÉ QUE ELES PASSEM .
- O IMPORTANTE É QUE NA INTEGRAÇÃO AS FUNCIONALIDADES SÓ PODEM SER INTEGRADAS SE NÃO HOUVER ERROS, CASO CONTRÁRIO OS ERROS DEVEM SER CORRIGIDOS.

SEMANA DE QUARENTA HORAS (40 HOUR WEEK)

- TRABALHAR POR LONGOS PERÍODOS É CONTRAPRODUCENTE. PORTANTO, SEMPRE QUE POSSÍVEL, DEVE-SE EVITAR A SOBRECARGA DE TRABALHO DE TODOS DA EQUIPE, CRIANDO CONDIÇÕES FAVORÁVEIS AO USO DA CARGA NORMAL DE TRABALHO.
- É NECESSÁRIO DEIXAR A EQUIPE LIVRE PARA RELAXAR, BRINCAR, OU FAZER O QUEM BEM ENTENDER PARA EQUILIBRAR O TRABALHO MENTAL E FÍSICO.

CLIENTE NO LOCAL (THE CUSTOMER IS ALWAYS AVAILABLE)

- DEVE SER INCLUÍDO NA EQUIPE UMA PESSOA DA PARTE DO CLIENTE, QUE IRÁ USAR O SISTEMA, PARA TRABALHAR JUNTO COM OS OUTROS E RESPONDER AS PERGUNTAS E DÚVIDAS;
- MESMO QUE NÃO SEJA POSSÍVEL OBTER ALGUÉM DA PARTE DO CLIENTE DEVE-SE TER ALGUÉM QUE TENHA CONHECIMENTO SUFICIENTE DO NEGÓCIO PARA EXERCER ESTE PAPEL;
- O CLIENTE TEM UM PAPEL IMPORTANTE DENTRO DE UM PROJETO XP JÁ QUE ELE PARTICIPA DO PLANEJAMENTO DO PROJETO ESCREVENDO AS HISTÓRIAS E PRIORIZANDO-AS.

GERÊNCIA DE PROJETOS E PAPÉIS ENVOLVIDOS EM XP

- A ESTRATÉGIA DE GERÊNCIA ADOTADA EM XP É MAIS VOLTADA PARA A TOMADA DE DECISÕES DESCENTRALIZADA DO QUE PARA O CONTROLE CENTRALIZADO.
- O PAPEL DO GERENTE É FAZER FLUIR O JOGO DO PLANEJAMENTO, COLETAR MÉTRICAS, FAZER COM QUE AS MÉTRICAS SEJAM VISTAS POR AQUELES CUJO TRABALHO ESTEJA SENDO MEDIDO, E OCASIONALMENTE INTERVIR EM SITUAÇÕES QUE NÃO PODEM SER RESOLVIDAS DE FORMA DISTRIBUÍDA.

GERÊNCIA DE PROJETOS E PAPÉIS ENVOLVIDOS EM XP

- PARA ENTENDERMOS O FUNCIONAMENTO PRECISAMOS CONHECER OS PRINCIPAIS PAPÉIS PRESENTES NO XP.

**GERENTE
DE
PROJETO**

**PROGRAMA-
DOR**

**TREINADOR
(COACH)**

**ACOMPANHA-
DOR
(TRACKER)**

CLIENTE

GERÊNCIA DE PROJETOS E PAPÉIS ENVOLVIDOS EM XP

- A GERÊNCIA EM XP É DIVIDIDA ATRAVÉS DE DOIS PAPÉIS: O TREINADOR (COACH) E O RASTREADOR (TRACKER). ESSES PAPÉIS PODEM OU NÃO SER EXECUTADOS PELA MESMA PESSOA.
 - **TREINADOR:** SE PREOCUPA PRINCIPALMENTE COM A EXECUÇÃO TÉCNICA E EVOLUÇÃO DO PROCESSO. O TREINADOR IDEAL DEVE SER UM BOM COMUNICADOR, TER UM BOM CONHECIMENTO TÉCNICO E SER CONFIANTE. O PAPEL DO TREINADOR NÃO É DE TOMAR DECISÕES TÉCNICAS, MAS DE FAZER COM QUE TODOS TOMEM BOAS DECISÕES E DE FACILITAR O PROCESSO DE DESENVOLVIMENTO.
 - **RASTREADOR:** COLETA MÉTRICAS SOBRE O QUE ESTÁ SENDO DESENVOLVIDO E CONFRONTA COM AS MÉTRICAS ESTIMADAS VERIFICANDO POSSÍVEIS DIVERGÊNCIAS. O RASTREADOR DEVE TOMAR CUIDADO PARA NÃO PERTURBAR MUITO OS PROGRAMADORES PEDINDO POR MÉTRICAS.

GERÊNCIA DE PROJETOS E PAPÉIS ENVOLVIDOS EM XP

- ALÉM DOS PAPÉIS GERENCIAIS APRESENTADOS ANTERIORMENTE, UMA EQUIPE QUE UTILIZA XP PARA DESENVOLVER SOFTWARE É COMPOSTA DE OUTROS PAPÉIS. ESTES SÃO: **PROGRAMADOR, CLIENTE, TESTADOR E CONSULTOR.**



GERÊNCIA DE PROJETOS E PAPÉIS ENVOLVIDOS EM XP

- **PROGRAMADOR:** OCUPA O PRINCIPAL PAPEL EM XP. ELE ANALISA, PROJETA, TESTA, CODIFICA, E INTEGRA O SISTEMA.
 - ALÉM DISSO, O PROGRAMADOR ESTIMA A DIFICULDADE DAS HISTÓRIAS, FAZ ALTERAÇÕES NESSAS ESTIMATIVAS, CASO NECESSÁRIO. EM XP O FOCO ESTÁ NA PROGRAMAÇÃO E O IMPORTANTE É ENTREGAR FUNCIONALIDADES IMPLEMENTADAS PARA O CLIENTE. O PROGRAMADOR ESTÁ SEMPRE ESCREVENDO TESTES DE UNIDADE, CODIFICANDO E FAZENDO REFATORAÇÃO COM O OBJETIVO DE PRODUZIR CÓDIGO DE ALTA QUALIDADE RAPIDAMENTE.

GERÊNCIA DE PROJETOS E PAPÉIS ENVOLVIDOS EM XP

- **CLIENTE:** ESCOLHE O QUE VAI AGREGAR VALOR AO SEU NEGÓCIO, ESCOLHE O QUE DEVE SER FEITO PRIMEIRO E O QUE DEVE SER ADIADO.
 - ALÉM DISSO, O CLIENTE DEFINE COM A AJUDA DOS TESTADORES, OS TESTES FUNCIONAIS QUE IRÃO MOSTRAR SE O SISTEMA REALMENTE FAZ O QUE DEVE SER FEITO.



GERÊNCIA DE PROJETOS E PAPÉIS ENVOLVIDOS EM XP

- **TESTADOR:** IRÁ AJUDAR O CLIENTE NA DEFINIÇÃO E ESCRITA DOS TESTES FUNCIONAIS. ELE NÃO PRECISA SER UMA PESSOA COM APENAS ESSA FUNÇÃO, PODE DESEMPENHAR TAMBÉM O PAPEL DE PROGRAMADOR.



GERÊNCIA DE PROJETOS E PAPÉIS ENVOLVIDOS EM XP

- **CONSULTOR:** É NECESSÁRIO APENAS EM ALGUMAS SITUAÇÕES ONDE SE PRECISA DE ALGUÉM COM UM ELEVADO NÍVEL DE CONHECIMENTO, POR EXEMPLO, UM ESPECIALISTA EM UMA DETERMINADA TECNOLOGIA SOBRE DETERMINADO ASSUNTO QUE NÃO ESTÁ SENDO BEM COMPREENDIDO PELAS PESSOAS DO PROJETO.

QUANDO NÃO SE DEVE USAR XP

- OS LIMITES EXATOS DE XP AINDA NÃO SÃO CONHECIDOS, MAS EXISTEM ALGUNS FATORES QUE SÃO FORTES INDICADORES PARA NÃO USAR XP COMO: **EQUIPES GRANDES, CLIENTES DESCONFIADOS E TECNOLOGIA QUE NÃO DÁ SUPORTE FACILITADO PARA MUDANÇAS.**

QUANDO NÃO SE DEVE USAR XP

- **CULTURA:** A ORGANIZAÇÃO PODE ESTAR INSERIDA DENTRO DE UMA CULTURA TRADICIONAL DE DESENVOLVIMENTO DE SOFTWARE, PRODUZINDO MUITA DOCUMENTAÇÃO, GASTANDO MUITO TEMPO COM ANÁLISE E PROJETO ANTECIPADO, ENTRE OUTRAS COISAS. FAZER COM QUE A EQUIPE PASSE A ADOPTAR AS PRÁTICAS DE XP E ESQUEÇA AS ANTIGAS PODE SER MUITO DIFÍCIL.

QUANDO NÃO SE DEVE USAR XP

- **TAMANHO DA EQUIPE:** UM PROJETO XP DEVE POSSUIR UMA EQUIPE PEQUENA – GERALMENTE ATÉ 12 PROGRAMADORES. É DIFÍCIL IMAGINAR COMO FICARIAM ALGUNS CONCEITOS DE XP COMO COMUNICAÇÃO, PROGRAMAÇÃO EM PAR E OUTRAS EM UMA EQUIPE GRANDE.

QUANDO NÃO SE DEVE USAR XP

- **ESPAÇO FÍSICO:** A ORGANIZAÇÃO DO ESPAÇO FÍSICO ONDE A EQUIPE DE XP TRABALHA DEVE FACILITAR A COMUNICAÇÃO E DEIXAR TODOS PRÓXIMOS UNS DOS OUTROS.

QUANDO NÃO SE DEVE USAR XP

- **CLIENTE:** XP EXIGE QUE O CLIENTE PARTICIPE DA EQUIPE DO PROJETO E TRABALHE NO MESMO LOCAL DOS DEMAIS, ESTANDO A DISPOSICÃO, DE PREFERÊNCIA, O TEMPO TODO PARA ESCLARECER DÚVIDAS.

XP NAS EMPRESAS ATUALMENTE

- A UTILIZAÇÃO DA XP JÁ É NOTÓRIA EM DIVERSAS EMPRESAS NO MUNDO.
- DESDE A SUA CRIAÇÃO E APLICAÇÃO NA EMPRESA CHRYSLER, INÚMERAS EMPRESAS AO REDOR DO MUNDO VÊM UTILIZANDO XP. ISSO MOSTRA QUE ESSA METODOLOGIA É MADURA, E APRESENTA UMA ALTERNATIVA NO DESENVOLVIMENTO DE SOFTWARE PARA ACOMPANHAR A TENDÊNCIA MUNDIAL DE DESENVOLVER COM AGILIDADE E DINAMISMO, COM FOCO PRINCIPALMENTE NA PRODUÇÃO DE CONHECIMENTO (TELES, 2005).

XP NAS EMPRESAS ATUALMENTE

Empresa	Website
37signals	http://37signals.com
Odeo	http://odeo.com
IBM	http://ibm.com
Interlegis	http://www.interlegis.gov.br
Royal Bank of Scotland	http://www.rbs.co.uk
Google	http://www.google.com
Objective Solutions	http://www.objectivesms.com.br/

Tabela 1: Empresas que utilizam XP em alguns de seus projetos

PRINCIPIOS

- RENEGA TODOS OS PARADIGMAS DO DESENVOLVIMENTO TRADICIONAL DE SOFTWARE;
- GANHOU DESTAQUE A PARTIR DA OOPLSA 2000 (A MAIOR CONFERÊNCIA INTERNACIONAL DE ORIENTAÇÃO A OBJETOS);
- A PRIMEIRA REAÇÃO, FOI BEM COMPLICADA POIS HOUVE UMA DIVISÃO ENTRE APOIADORES E CRÍTICOS QUE ACUSAVAM A METODOLOGIA DE AVACALHAMENTO DE REGRAS, O PROGRAMADOR FICAVA MUITO LIVRE.

CONCEITO

- SEGUNDO BECK (2000), XP É UMA METODOLOGIA ÁGIL PARA EQUIPES PEQUENAS E MÉDIAS (ATÉ 12 DESENVOLVEDORES), DESENVOLVENDO SOFTWARE COM REQUISITOS VAGOS OU QUE MUDAM FREQUENTEMENTE;
- PODE-SE DIZER TAMBÉM QUE, XP É UMA METODOLOGIA PARA DESENVOLVIMENTO DE SOFTWARE ÁGIL, COM QUALIDADE E QUE ATENDA AS NECESSIDADES DO CLIENTE. ALGUNS PRATICANTES DEFINEM A XP COMO A PRÁTICA E A PERSEGUIÇÃO DA MAIS CLARA SIMPLICIDADE, APLICADO AO DESENVOLVIMENTO DE SOFTWARE;
- A XP BUSCA O MÁXIMO DE VALOR A CADA DIA DE TRABALHO DA EQUIPE PARA O SEU CLIENTE (ENTREGAS EM CURTOS PRAZOS) PROPONDO ACOMPANHAMENTO E REAVALIAÇÃO CONSTANTE DO CÓDIGO E DO PROJETO. ESTANDO ABERTO A NOVOS AJUSTES QUANDO NECESSÁRIO.

EXEMPLO DE APLICAÇÃO

- FORD MOTORS COMPANY – PROJETO DE DESENVOLVIMENTO DO VCAPS SYSTEM, QUE ERA UM SISTEMA DE CÁLCULO DE CUSTO DE UM VEÍCULO. USANDO METODOLOGIAS TRADICIONAIS ELES PASSARAM QUARTO ANOS E NÃO TINHAM FEITO NEM 30% DO PROJETO. APÓS ADOPTAR TESTES AUTOMATIZADOS, INTEGRAÇÃO CONTÍNUA E OUTROS VALORES DO XP, EM UM ANO, O SISTEMA FOI DESENVOLVIDO E EVOLUÍDO CONSTANTEMENTE SEM MAIORES PROBLEMAS.

EXEMPLO DE APLICAÇÃO

- CHRYSLER – UTILIZADO NO DESENVOLVIMENTO DO SEU SISTEMA DE FOLHA DE PAGAMENTO GLOBAL QUE ENVOLVIA SEUS 90 MIL EMPREGADOS AO REDOR DE TODO O MUNDO. ESSE PROJETO TEM 2.000 CLASSES E 30.000 MÉTODOS, E FOI PARA PRODUÇÃO SEM NENHUM ATRASO.

CHRYSLER



SCRUM X KABAN X XP

	Scrum	Kanban	XP
Objetivo	Uso de equipes multifuncionais, auto organizadas e capacitadas que dividem seu trabalho em curtos e concentrados ciclos de trabalho chamados Sprints.	Para aliviar os impedimentos que nos levam a levar mais tempo para entregar, não remover peças necessárias do processo.	Organizar as pessoas para produzir software de qualidade superior de forma mais produtiva.
Funções	Funções principais: Product Owner, Scrum Master e Scrum Team (Equipe)	Nenhuma função existente. Algumas equipes exigem a ajuda de um treinador ágil.	Acompanhador (Tracker), Cliente, Programador, Treinador (coach), Gerente de projeto.

SCRUM X KANBAN X XP

	Scrum	Kanban	XP
Principais métricas	Velocidade de Sprint (2 semanas).	Tempo de ciclo.	Tempo de iteração (2 semanas).
Mudança de filosofia	As equipes devem se esforçar para não fazer alterações na previsão de sprint durante a sprint. Isso compromete as aprendizagens em torno da estimativa.	A mudança pode acontecer a qualquer momento.	Um alto grau de disciplina de desenvolvedores junto com o envolvimento contínuo do cliente durante todo o projeto.
Cadência	Sprints regulares de comprimento fixo.	Fluxo contínuo.	Iteração.
Metodologia de liberação	No final de cada sprint se aprovado pelo proprietário do produto.	Entrega contínua ou a critério da equipe.	No final da iteração.

