

ENGENHARIA DE SOFTWARE



Aula 3

Professor.: Carlos Danilo



Tópicos da aula

- Metodologias Ágeis;
- Scrum
- Kanban
- Extreme Programming (XP);



Metodologias Ágeis

- Os métodos ágeis são uma abordagem ao modelo de gestão tradicional de projetos.
- O modelo de entrega ágil é baseado em ciclos iterativos e incrementais, o que traz flexibilidade e adaptabilidade.
- Uma característica importante é a inspeção e adaptação dos ciclos e iterações, focados em gerar melhoria contínua para as equipes e processos.
- Estão focados no que agrega valor ao cliente.



Um pouco da história...

O que é Taylorismo?

O Taylorismo, desenvolvido pelo americano Frederick Winslow Taylor, é uma teoria administrativa focada na racionalização do trabalho para elevar a produtividade.

- Divisão do trabalho em tarefas específicas;
- Aumento da produtividade;
- Grande nível de subordinação.



Um pouco da história...

O que é Fordismo?

O Fordismo, criado por Henry Ford em 1908, é uma evolução do Taylorismo. Embora tenha preservado o sistema anterior, introduziu a esteira rolante, promovendo um ritmo de trabalho mais ágil e eficiente.

- Padronização dos produtos;
- Produção em grande escala;
- Uso de linhas de montagem;
- Divisão do trabalho em pequenas tarefas.



Um pouco da história...

Tempos Modernos - <https://www.youtube.com/watch?v=4yl2L6pdSYg>



Um pouco da história...

O que é Toyotismo?

O toyotismo, desenvolvido por Taiichi Ohno em 1962 na Toyota, é um modelo de organização do trabalho que se baseia em dois princípios fundamentais:

- Princípio Just in Time (JIT): visa reduzir estoques ao produzir conforme a demanda.
- Princípio dos Cinco Zeros: alcançar zero atraso, zero defeitos, zero estoque, zero falhas e zero papel.



Um pouco da história...

O que é Toyotismo?

- Produção diversificada;
- Eliminação de desperdício;
- Autonomia;
- Trabalhadores com múltiplas tarefas.



Um pouco da história...

O que é Toyotismo?

Após a Segunda Guerra Mundial, a Toyota desenvolveu o Toyota Production System (TPS), também conhecido como Lean Manufacturing. Este sistema inovador visa a produção enxuta, concentrando-se na redução de custos por meio da eliminação de desperdícios.



Um pouco da história...

O que é Toyotismo?

Muitas pessoas ligam a origem dos Métodos Ágeis com a abordagem da **Toyota** e ao **Lean**.

Outros ligam ao artigo *The New New Product Development Game* escrito por **Hiroataka Takeuchi** e **Ikujiro Nonaka**, onde os autores abordam produção iterativa.



O Manifesto Ágil

Em 2001 um grupo de 17 pessoas se reuniram para discutir sobre uma nova abordagem para a gestão de projetos de software. Ao final dessa reunião todas as pessoas presentes assinaram o que é conhecido como Manifesto Ágil.

[Manifesto para Desenvolvimento Ágil de Software](#)



O Manifesto Ágil

O manifesto é composto por **Quatro Pilares**

Indivíduos e interações mais que processos e ferramentas.

Software em funcionamento mais que documentação abrangente.

Colaboração com o cliente mais que negociação de contratos.

Responder a mudanças mais que seguir o plano.

Ou seja mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.



O Manifesto Ágil

Princípios da Agilidade

- 1 - “Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado”
- 2 - “Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente”



0 Manifesto Ágil

Princípios da Agilidade

3 - “Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo”

4 - “Pessoas de negócio e desenvolvedores devem trabalhar em conjunto diariamente por todo o projeto”



O Manifesto Ágil

Princípios da Agilidade

5 - “Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho”

6 - “O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é por meio de conversa face a face”



O Manifesto Ágil

Princípios da Agilidade

7 - “Software funcionando é a medida primária de progresso”

8 - “Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente”



O Manifesto Ágil

Princípios da Agilidade

9 - “Contínua atenção à excelência técnica e bom design aumenta a agilidade”

10 - “Simplicidade – a arte de maximizar a quantidade de trabalho não realizado – é essencial”



O Manifesto Ágil

Princípios da Agilidade

11 - “As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis”

12 - “Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.”



Metodologias Ágeis

- Scrum
- Extreme Programming (XP)
- Lean Development
- Feature-Driven Development (FDD)
- Kanban
- Crystal
- Entre outras...

Metodologias Ágeis

Vantagens Clientes

Foco e maximização do ROI

Foco no que gera mais valor para o usuário

Redução dos riscos e indesejáveis surpresas

Escopo e objetivos claros e priorizados

Metodologias Ágeis

Vantagens Clientes

Transparência e visibilidade do status do projeto

Flexibilidade para mudanças, maior agilidade na tomada de decisões

Melhoria da qualidade do produto final

Metodologias Ágeis

Vantagens Gestor e Equipe

Equipes auto gerenciáveis, maior autonomia, disciplina e regularidade

Pessoas comprometidas

Melhoria na comunicação

Metodologias Ágeis

Vantagens Gestor e Equipe

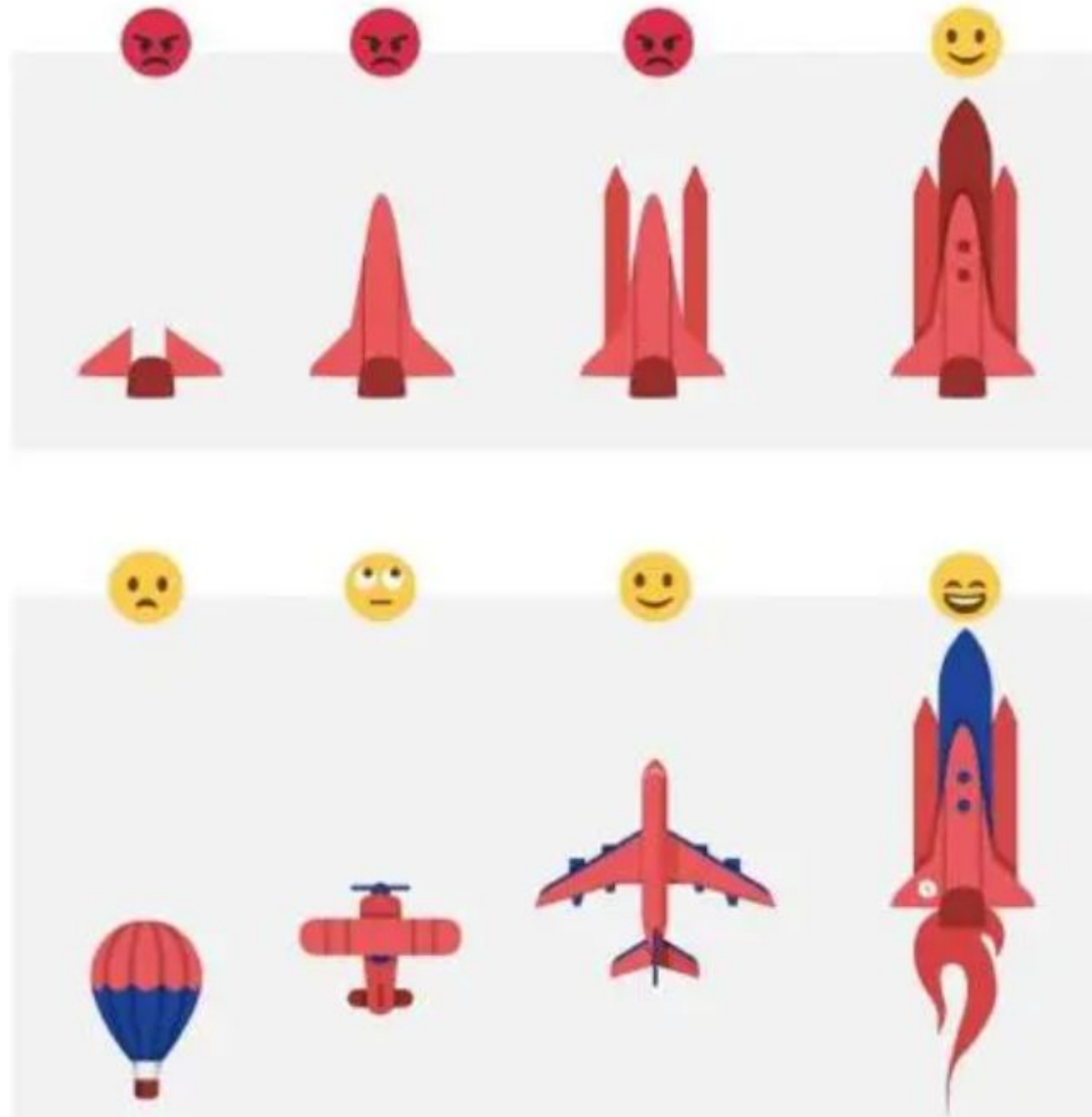
Inspeção e Adaptação constantes, busca da melhoria contínua

Antecipação dos problemas, agilidade na tomada de ações

Metodologias Ágeis

Foco MVP

(Minimum Viable Product)



Créditos da imagem - DB1 Global Software



Metodologias Ágeis

Características Ágeis

Envolvimento real do usuário final

Empoderamento da equipe

Uma peça de cada vez

Entregue pouco mas entregue sempre

Pense simples

Qualidade no que faz

Medir tudo o que pode ser medido

Automatize as coisas chatas

Kanban



Kanban

Metodologia visual de gerenciamento de tarefas que usa um quadro para representar o fluxo de trabalho.

As **tarefas são divididas em categorias**, como "**A Fazer**", "**Em Andamento**" e "**Concluído**", e movidas entre essas categorias conforme seu progresso.

O objetivo é visualizar o trabalho, identificar gargalos e melhorar a eficiência do processo.

Kanban – Papéis Comuns

Equipe de Desenvolvimento: Implementar, testar e concluir as tarefas

Gerente de Projetos ou Líder de Equipe: Monitorar o fluxo de trabalho, ajustar limites de WIP (Work In Progress), e promover a melhoria contínua.

Proprietário do Produto (ou Product Owner): Gerenciar o backlog, definir prioridades

Scrum Master, Facilitador, Agile Master: Ajudar a equipe a seguir o processo Kanban e resolver problemas que possam surgir.

Stakeholders: Revisar o progresso, dar feedback sobre as entregas e colaborar no planejamento de prioridades.



Kanban – Vantagens

Visualização Clara: Torna o fluxo de trabalho visível, facilitando a identificação de gargalos e áreas de melhoria.

Flexibilidade: Adaptável a diferentes tipos de projetos e equipes.

Melhoria Contínua: Promove a análise constante e a otimização do processo de trabalho.

Redução de Tempo de Ciclo: Ajuda a reduzir o tempo necessário para concluir tarefas ao identificar e eliminar desperdícios.



Kanban – Desvantagens

Escalabilidade: Pode ser menos eficaz para projetos muito grandes ou complexos com múltiplas equipes.

Dependência de Disciplina: Requer comprometimento da equipe para atualizar o quadro e seguir as práticas recomendadas.

Visibilidade Limitada: Pode não fornecer uma visão completa de prazos e dependências se não for integrado com outras ferramentas de gestão.

Kanban - Jira

Jira

Your work

Projects

Filter

Dashboards

Teams

Plans

Apps

Create

Search

?

Beyond Gravity

Software project

PLANNING

Roadmap

Backlog

Board

Reports

Issues

DEVELOPMENT

Code

Security

Releases

Project settings

Projects / Beyond Gravity

Board

+3

Epic

GROUP BY

None

Insights

TO DO 6

Optimize experience for mobile web

BILLING

NUC-344

Onboard workout options (OWO)

ACCOUNTS

NUC-360

Multi-dest search UI mobileweb

ACCOUNTS

NUC-337

Billing system integration - frontend

FORMS

NUC-339

Account settings defaults

ACCOUNTS

NUC-340

IN PROGRESS 6

Fast trip search

ACCOUNTS

NUC-342

Affelite links integration - frontend

BILLING

NUC-335

Shopping cart purchasing error - quick fix required

FORMS

NUC-341

IN REVIEW 6

Revise and streamline booking flow

ACCOUNTS

NUC-367

Travel suggestion experiments

ACCOUNTS

NUC-358

Ongoing customer satisfaction

ACCOUNTS

NUC-354

Planet Taxi Device exploration & research

FEEDBACK

NUC-351

DONE 6

High outage: Software bug fix - BG Web-store app crashing

BILLING

NUC-340

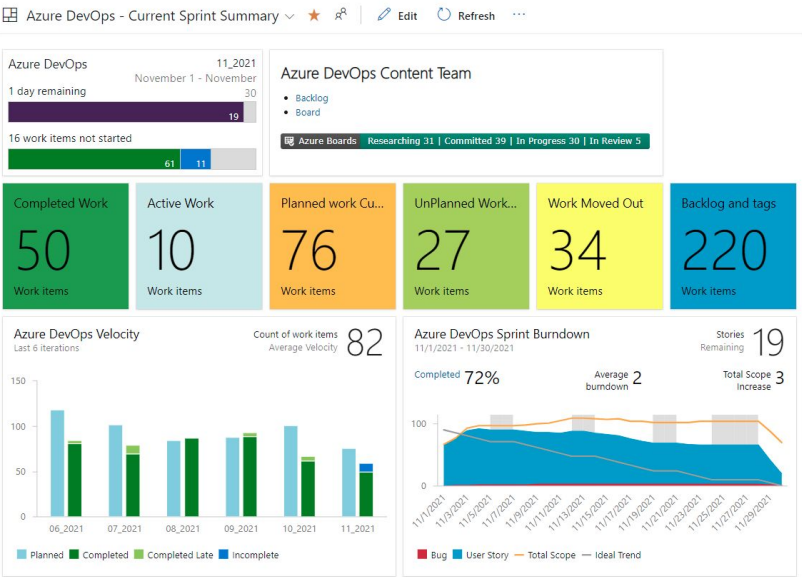
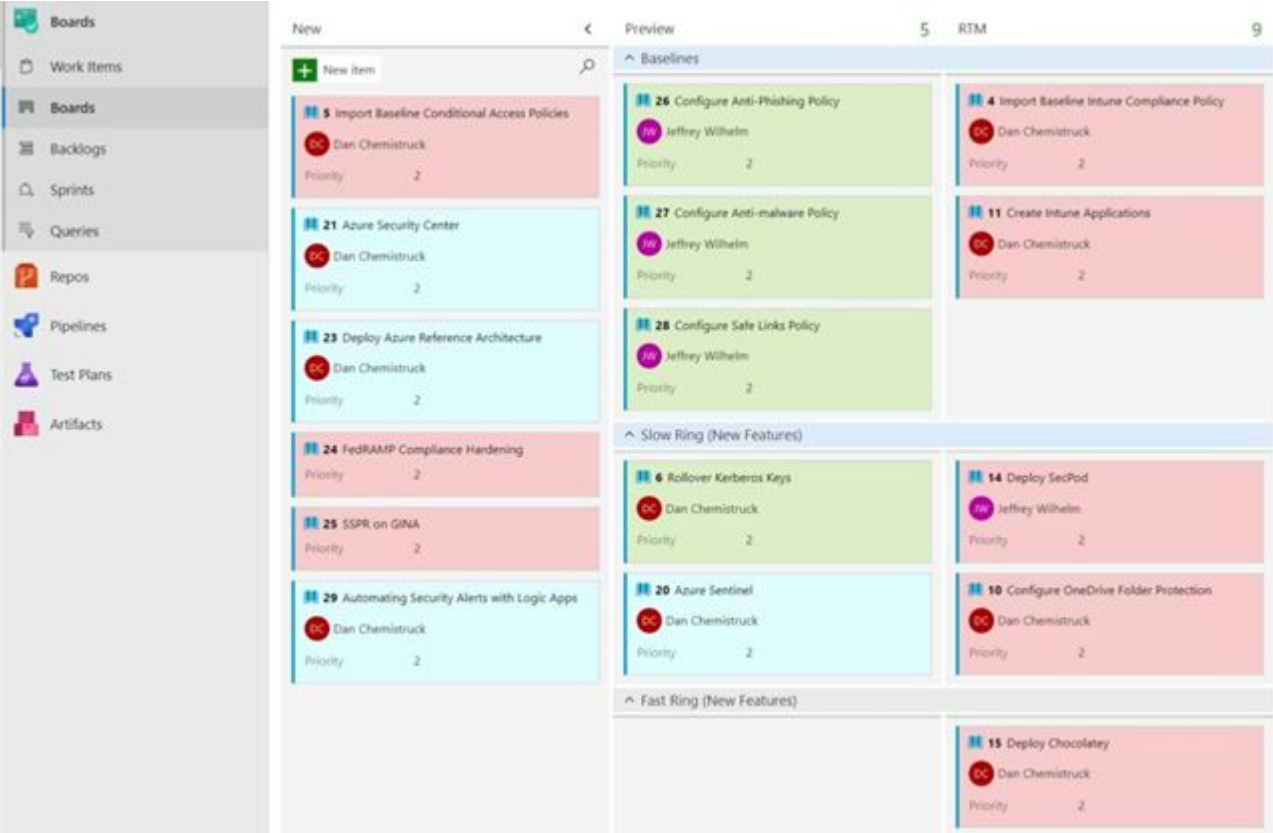
Web-store purchasing performance issue fix....t

FORMS

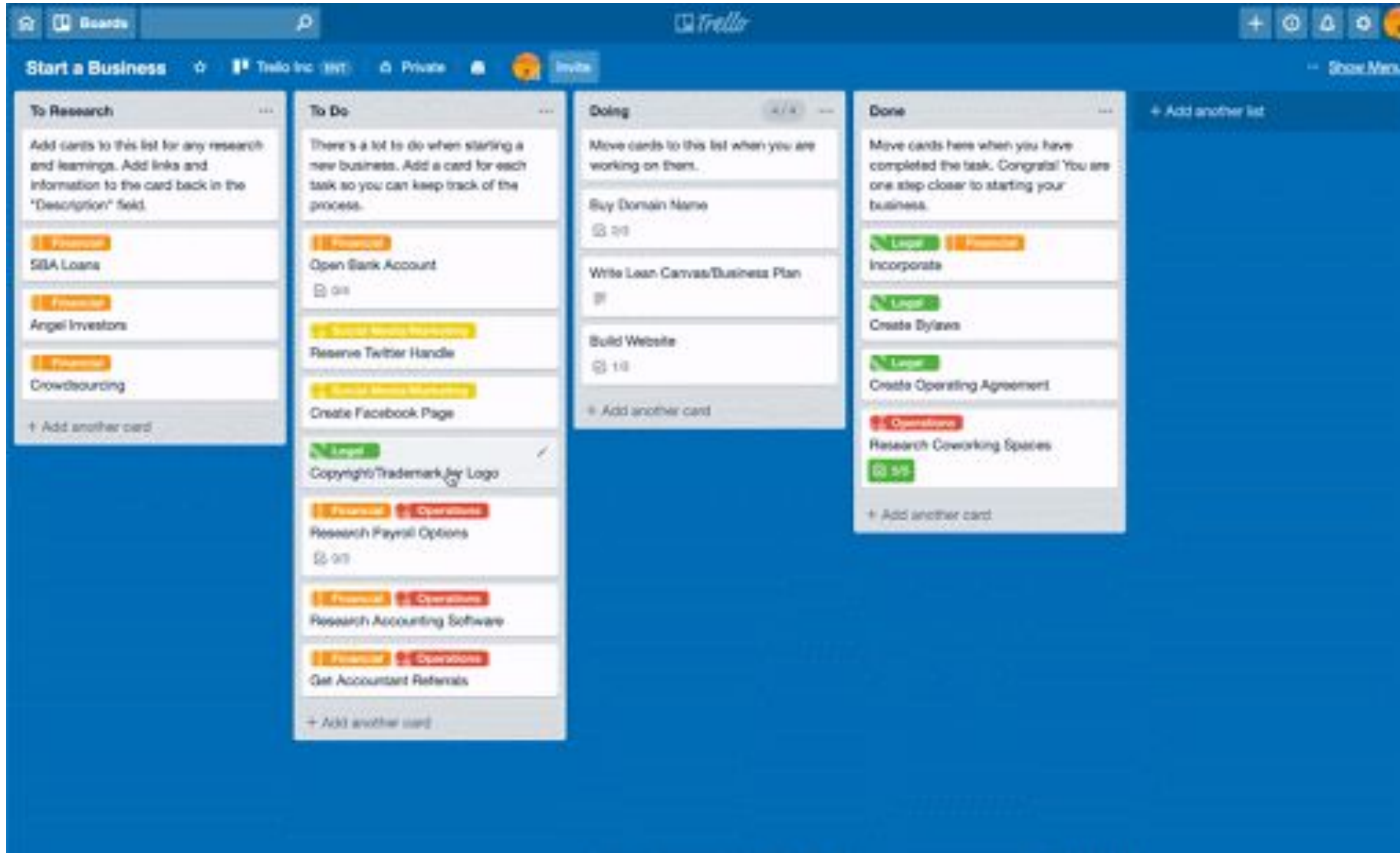
NUC-341

#2022REALIZAR

Kanban – Azure Boards



Kanban – Trello



Kanban

Integração com Ferramentas: Várias ferramentas digitais de Kanban, como Trello, Jira, Asana, Azure que permitem o gerenciamento remoto e colaborativo.

Adaptação ao Contexto: É importante ajustar o método Kanban às necessidades específicas da equipe e do projeto, como ajustar limites de WIP ou adicionar colunas personalizadas.




```
header1.css('padding-top', 10) += header1_initialPadding + $(window).scrollTop() - header1_initialDistance) {\n    }\n} else {\n    header1.css('padding-top', '' + header1_initialPadding + 'px');\n}\n\nif ($(window).scrollTop() > header2_initialDistance) {\n    if (parseInt(header2.css('padding-top'), 10) += header2_initialPadding + $(window).scrollTop() - header2_initialDistance) {\n        header2.css('padding-top', '' + $(window).scrollTop() - header2_initialDistance + 'px');\n    }\n}
```

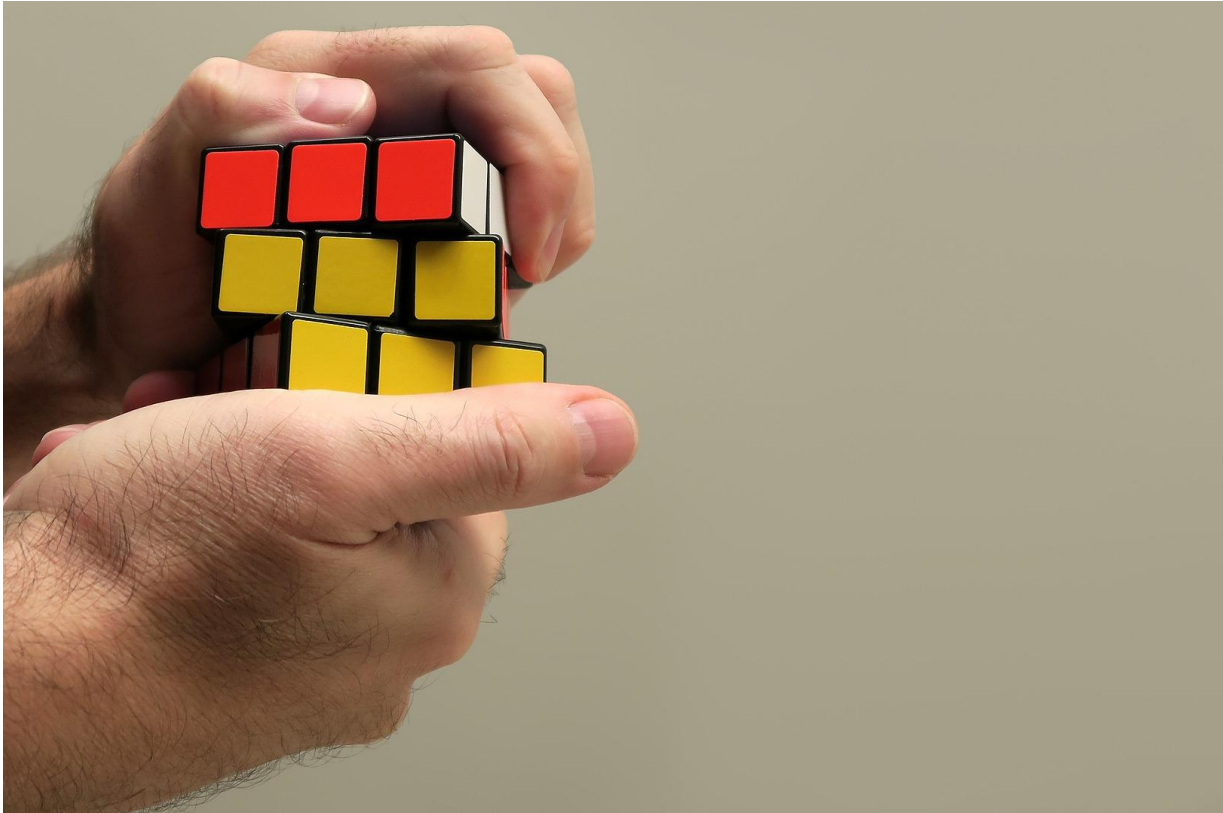
Extreme Programming (XP)

Um conjunto de práticas, valores e princípios que levam as boas práticas de programação ao seu máximo potencial.

XP em uma palavra: **Qualidade**



Extreme Programming (XP)



O que resolve?

- Especificação mal apresentada;
- O famoso: tá pronto, só falta testar!

Extreme Programming (XP)

- **Leve:** Porque os processos vão ser mantidos ao mínimo;
- **Humana:** Destaca o papel essencial do desenvolvedor, do cliente e de todos os colaboradores no projeto.
- **Disciplinada:** Porque adota um conjunto de valores, práticas e princípios.

Extreme Programming (XP)

Criada por Kent Beck nos anos 1990 renomado desenvolvedor de software, um dos 17 signatários do Manifesto Ágil. Beck desenvolveu XP enquanto trabalhava em um projeto na Chrysler, onde procurava uma maneira de melhorar a eficiência e a qualidade do desenvolvimento de software.

Tem ênfase na melhoria contínua, comunicação constante, feedback rápido, e um forte foco na qualidade do código.



Extreme Programming (XP) – Práticas

Programação em Par: Dois desenvolvedores trabalham juntos no mesmo código, alternando entre quem escreve o código e quem revisa.

Desenvolvimento Orientado a Testes (TDD): Escrever testes antes do código para garantir que o software funciona conforme esperado.

Integração Contínua: Código é integrado e testado várias vezes ao dia para identificar problemas rapidamente.

Refatoração: Melhorar continuamente o código para torná-lo mais simples e mais fácil de manter.

Pequenas Releases: Entregar software funcional em pequenos incrementos para obter feedback rápido dos usuários.



Extreme Programming (XP) - Valores

Comunicação: Sua ausência pode causar retrabalho e complicar a execução do projeto.

Simplicidade: Procure a solução mais simples e eficiente. Evite reinventar a roda.

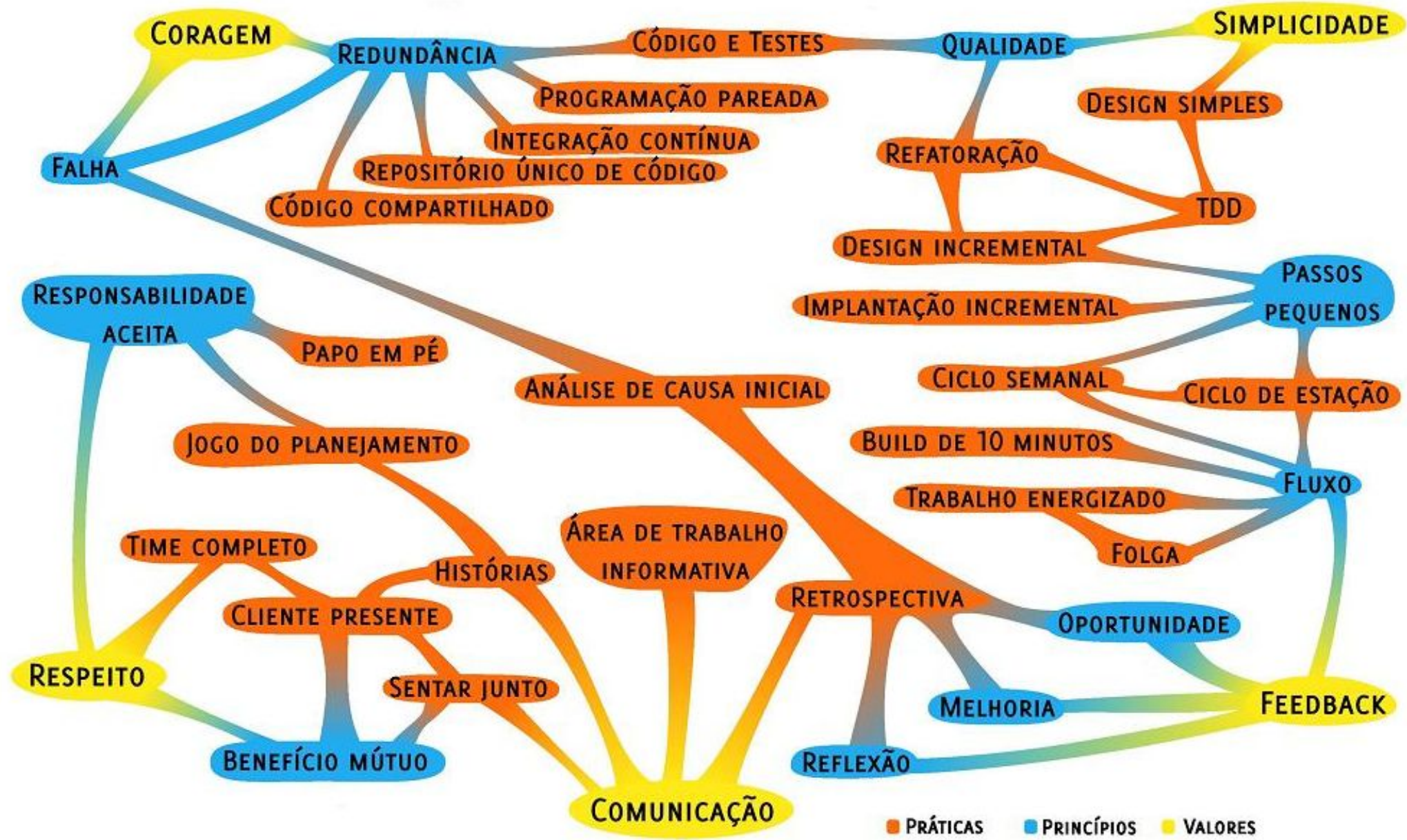
Feedback: Essencial para o processo, pois o retorno do cliente é crucial para realizar os ajustes necessários;

Coragem: No processo de feedback é necessário ter coragem para descartar o código e recomeçar o projeto do zero.

Respeito: Entre o time, para o benefício geral do time e do cliente;

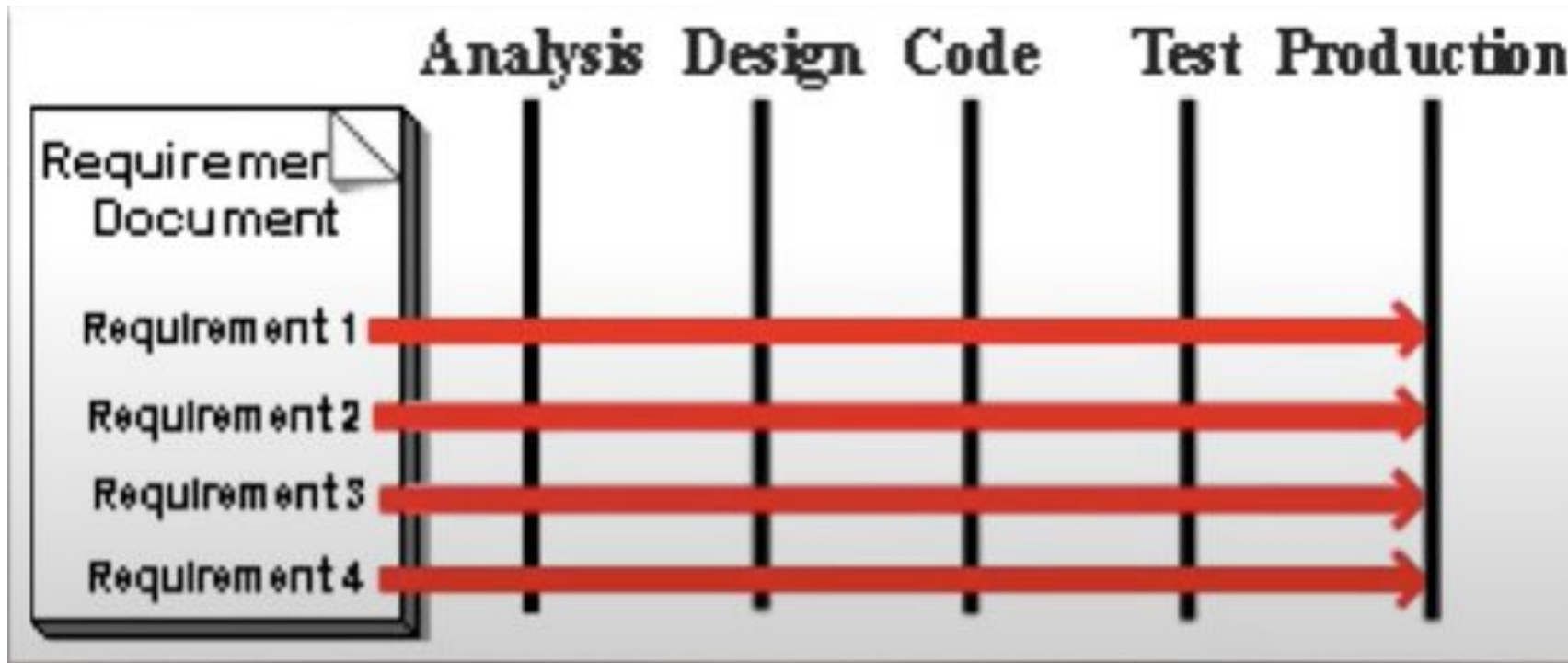


Extreme Programming (XP)



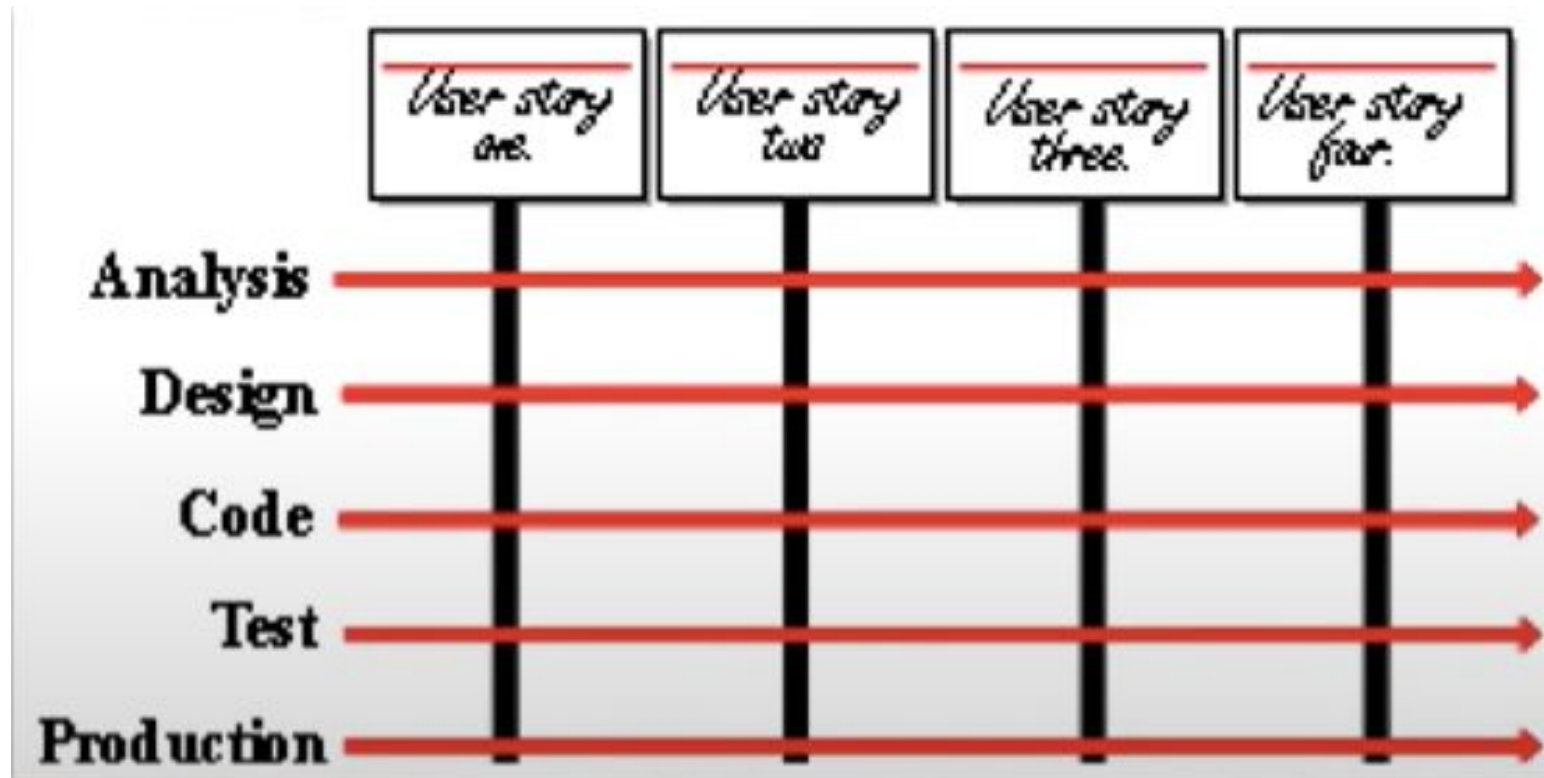
Extreme Programming (XP)

Ciclo tradicional dos requisitos



Extreme Programming (XP)

XP sugere Storys



Extreme Programming (XP)

XP sugere Storys

- User Stories são descrições simples e curtas de uma funcionalidade do ponto de vista do usuário.
- São usadas em metodologias ágeis (Scrum, XP, Kanban) para capturar necessidades e expectativas.
- Devem focar no valor entregue ao usuário e não em detalhes técnicos.

Extreme Programming (XP)

XP sugere Storys

- Linguagem simples e acessível.
- Escritas em primeira pessoa.
- Seguem geralmente o formato:

“Como [tipo de usuário], quero [funcionalidade] para [benefício/valor]”.

Extreme Programming (XP)

XP sugere Storys

- "Como cliente do banco, eu quero visualizar meu saldo em tempo real para acompanhar minhas finanças."
- Critérios de Aceitação
 - O saldo deve ser atualizado automaticamente após transações.
 - O usuário deve conseguir visualizar o saldo na tela inicial do app.
 - O sistema deve indicar data e hora da última atualização.

Extreme Programming (XP)

Em XP o gerente

- Não define prioridades; o cliente faz isto.
- Não delega tarefas nem duração das tarefas; os programadores fazem isto.
- Não definem cronogramas; clientes e programadores fazem isto.

Extreme Programming (XP)

Em XP, o papel do gerente é menos centralizado em controle e mais focado em apoiar a equipe e facilitar o processo de desenvolvimento.

Tá, mas o que eles faz então?



Extreme Programming (XP)

Facilitador: Remove obstáculos e impedimentos.

Coordenador: Facilita a comunicação entre o desenvolvimento e outras parte.

Apoiador do Cliente: Garante que o cliente esteja engajado e disponível para fornecer feedback.

Gerenciador de Recursos: Assegura os recursos e ferramentas para trabalhar de forma eficiente.

Promotor da Cultura de XP: Incentiva a adoção e a prática dos princípios

Tomador de Decisões Estratégicas: Decisões sobre metas e prioridades do projeto.

Monitoramento e Ajustes: Acompanha o progresso do projeto e faz ajustes.



Dúvidas ou perguntas?