

Scrum e Programação Extrema (XP)

Livro de referência: Programação eXtrema eXplicada, por Kent Beck

documento por Gustavo Choueiri

1. Introdução

Scrum e Programação Extrema (XP) são metodologias ágeis, mas com focos distintos:

- **Scrum:** Framework de **gestão de projetos** para problemas complexos, com ênfase em **iterações curtas (Sprints)** e **auto-organização**.
- **XP:** Metodologia de **desenvolvimento técnico**, centrada em **qualidade do código**, **feedback contínuo** e **práticas de engenharia rigorosas**.

Objetivo deste documento:

- Explicar os princípios de cada metodologia.
- Mostrar como se complementam.
- Identificar pontos de atenção (como riscos e más interpretações).

2. Programação **Extrema** (XP)

2.1 O Problema Básico no Desenvolvimento de Software

A XP surge para resolver **riscos crônicos** em projetos tradicionais:

1. **Deslize no cronograma:** Prazos irreais ou falta de adaptação a mudanças.
 - a. *Exemplo:* Um projeto com "deadline" fixo, mas requisitos não consolidados, gera retrabalho.
2. **Falsa riqueza de funcionalidades:** Features "bonitas" mas inúteis para o usuário final.
 - a. *Exemplo:* Um sistema com 10 telas de configuração, mas que não resolve o problema central do cliente.
3. **Taxa de erros alta:** Falta de testes ou feedback tardio.
 - a. **Solução XP:** Testes automatizados desde o início (TDD) e pair programming.

Regras da XP para mitigar riscos:

- **"Diga SIM ao pedido de ajuda"**: Evita gargalos e promove colaboração.
- **"Terminou quando os testes passam"**: Funcionalidade só é concluída após cobertura total de testes.

2.2 As 4 Variáveis do Projeto (XP)

A XP propõe equilibrar quatro variáveis interdependentes:

Variável	Explicação	Analogia/Exemplo
Escopo	Menor escopo = maior qualidade e entrega rápida.	"9 mulheres não fazem 1 bebê em 1 mês": Adicionar mais pessoas não acelera tarefas indivisíveis.
Tempo	Controlado pelo cliente (não pelo gerente). No Scrum, equivale ao Product Owner .	Exemplo: Cliente prioriza entre "entregar logo" ou "adicionar mais funcionalidades".
Qualidade	Qualidade interna (código) impacta a externa (produto). Sacrificá-la gera dívida técnica .	Exemplo: Pular refatoração para cumprir prazo = bugs futuros exponenciais.
Custo	Recursos adicionais (ex.: mais devs) nem sempre aceleram o projeto.	Lei de Brooks: "Adicionar pessoas a um projeto atrasado o atrasa mais".

Princípio central: Todas as variáveis devem ser **visíveis** para a equipe tomar decisões conscientes.

2.3 Os 4 Valores da XP

1. Comunicação

- a. **Evitar silos de informação**. Práticas como **pair programming** e reuniões diárias (similares ao Daily Scrum).

2. Simplicidade

- "Faça o mais simples que funciona hoje" vs. superengenharia antecipada.
- Exemplo: Criar um módulo básico de login e iterar (em vez de planejar integração com 5 redes sociais de uma vez).

3. Feedback

- Testes automatizados e revisões contínuas.
- Frase-chave do documento original: "Muitas horas de discussão são poupadas se um colega mostrar um teste que falha no seu código."

4. Coragem

- Refatorar código mesmo sob pressão e admitir erros rapidamente.

Práticas técnicas da XP:

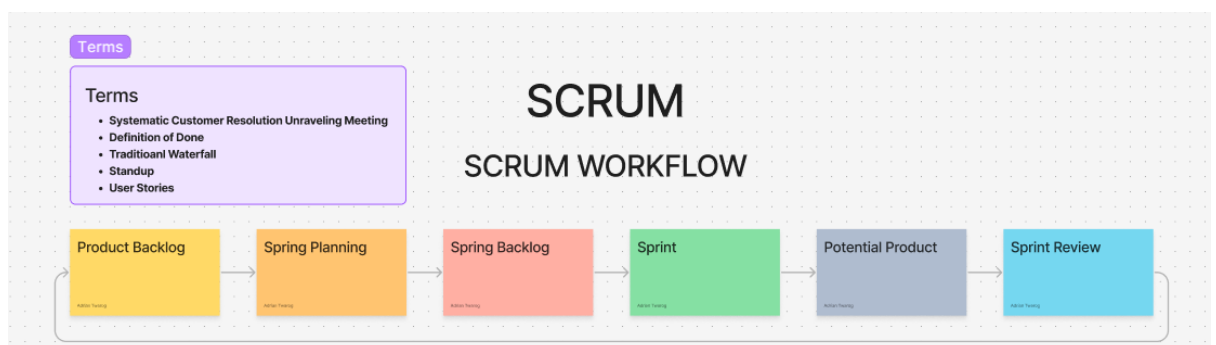
- TDD (Test-Driven Development): Escrever testes antes do código.

3. Scrum: Teoria e Prática

3.1 Definição e Princípios

Scrum é um **framework empírico** baseado em:

- Transparência**: Todos os artefatos (Backlog, Sprint) devem ser visíveis.
- Inspeção**: Revisões frequentes (ex.: Sprint Review).
- Adaptação**: Ajustar o plano após feedback (ex.: Retrospectiva).



Diferença chave para XP:

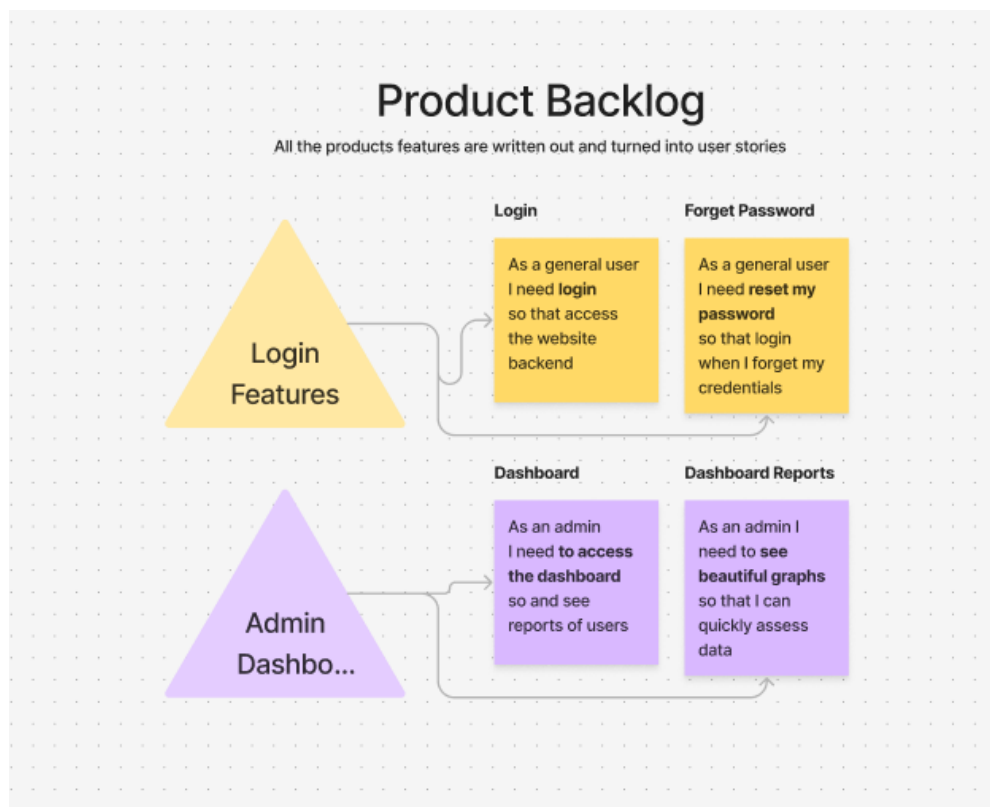
- Scrum não prescreve práticas técnicas (como TDD), apenas a **estrutura de gestão**.

3.2 O Time Scrum

Papel	Responsabilidades	Conexão com XP
Product Owner (PO)	Define prioridades do backlog e garante valor.	Similar ao "cliente presente" na XP, mas com mais autoridade.
Scrum Master	Remove impedimentos e ensina Scrum.	Equivalente ao "coach" de XP, mas focado no processo.
Developers	Autogerenciáveis, entregam incrementos.	Podem adotar práticas XP (como pair programming).

Frase-chave do documento original:

"Scrum Teams são auto-gerenciáveis: decidem internamente quem faz o quê, sem hierarquias."



Repare na foto acima, que ao selecionar elementos do "BACKLOG" (que é o que o produto irá ter, todas as funcionalidades, etc), é pego pequenas partes do produto backlog e então é adicionado a sprint.

Repare que as "frases formadas" se chamam "histórias" que são pequenos resumos sobre o que será feito e para que será feito.

3.3 Eventos do Scrum

1. Sprint

- a. Ciclo fixo (1-4 semanas) para entregar um incremento.
 - b. *Regra rígida*: **Nenhuma mudança no escopo durante a Sprint** (exceto em casos extremos).
2. **Daily Scrum**
 - a. Reunião de 15 minutos para sincronização (não é "relatório de status").
 3. **Sprint Review**
 - a. Mostrar o incremento ao cliente para feedback.
 4. **Retrospectiva**
 - a. Adaptar o processo (ex.: adotar TDD se a qualidade estiver baixa).

Conexão com XP:

- O feedback da **Sprint Review** alinha-se ao valor **Feedback** da XP.
- A **Retrospectiva** pode incluir práticas XP (ex.: "vamos tentar pair programming na próxima Sprint").

4. Scrum + XP: Como se Complementam

4.1 Exemplo Prático

Um time usando **Scrum** pode adotar **práticas da XP** para melhorar a qualidade:

- **Sprint Planning**: Selecionar user stories e já definir testes (TDD).
- **Durante a Sprint**: Pair programming para tarefas complexas.
- **Sprint Review**: Mostrar não só funcionalidades, mas **métricas de qualidade** (ex.: cobertura de testes).

4.2 Riscos de Não Combiná-las

- **Scrum sem XP**: Pode virar "Scrum de papel" (reuniões sem qualidade técnica).
- **XP sem Scrum**: Falta de estrutura para priorização e entrega contínua.

Imagens sobre SCRUM

