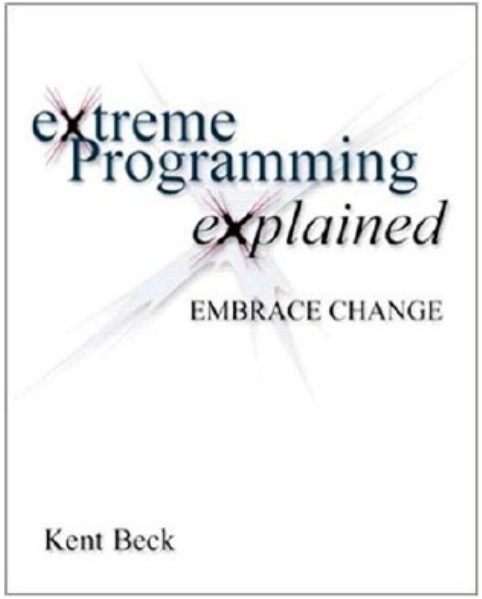
Modelos, Métodos e Técnicas de Engenharia de Software

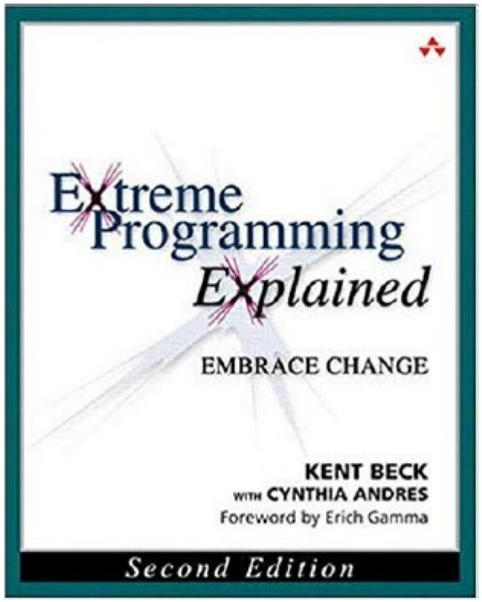
Prof.: Sônia A. Santana

una

Kent Beck



1999



2004

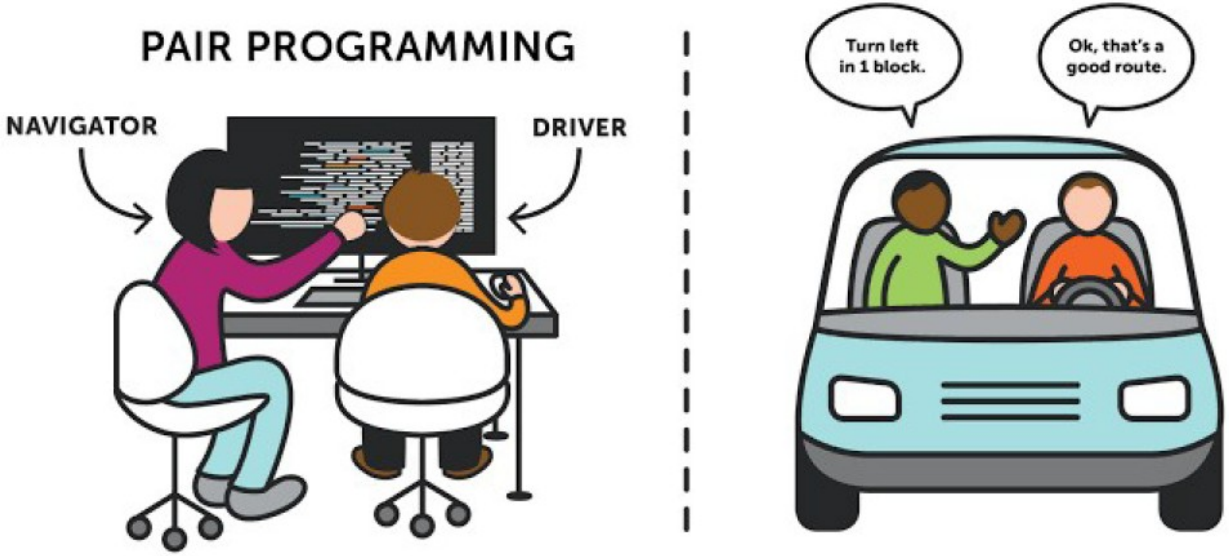
XP = Valores + Princípios + Práticas

- Comunicação
- Simplicidade
- Feedback
- Coragem
- Respeito

XP = Valores + Princípios + Práticas

- Economicidade
- Melhorias Contínuas
- Falhas Acontecem
- Baby Steps
- Responsabilidade Pessoal

XP = Valores + Princípios + Práticas



XP = Valores + Princípios + Práticas

- Jogo de Planejamento (Planning Game)
- Fases pequenas (Small Releases)
- Metáfora (Metaphor)
- Design Simples (Simple Design)
- Testes de Aceitação (Customer Tests)
- Ritmo Sustentável (Sustainable Pace)
- Propriedade Coletiva (Collective Ownership)
- Programação em Pares (Pair Programming)
- Padronização do Código (Coding Standards)
- Desenvolvimento Orientado a Testes (Test Driven Development)
- Refatoração (Refactoring)

Extreme Programming (XP)

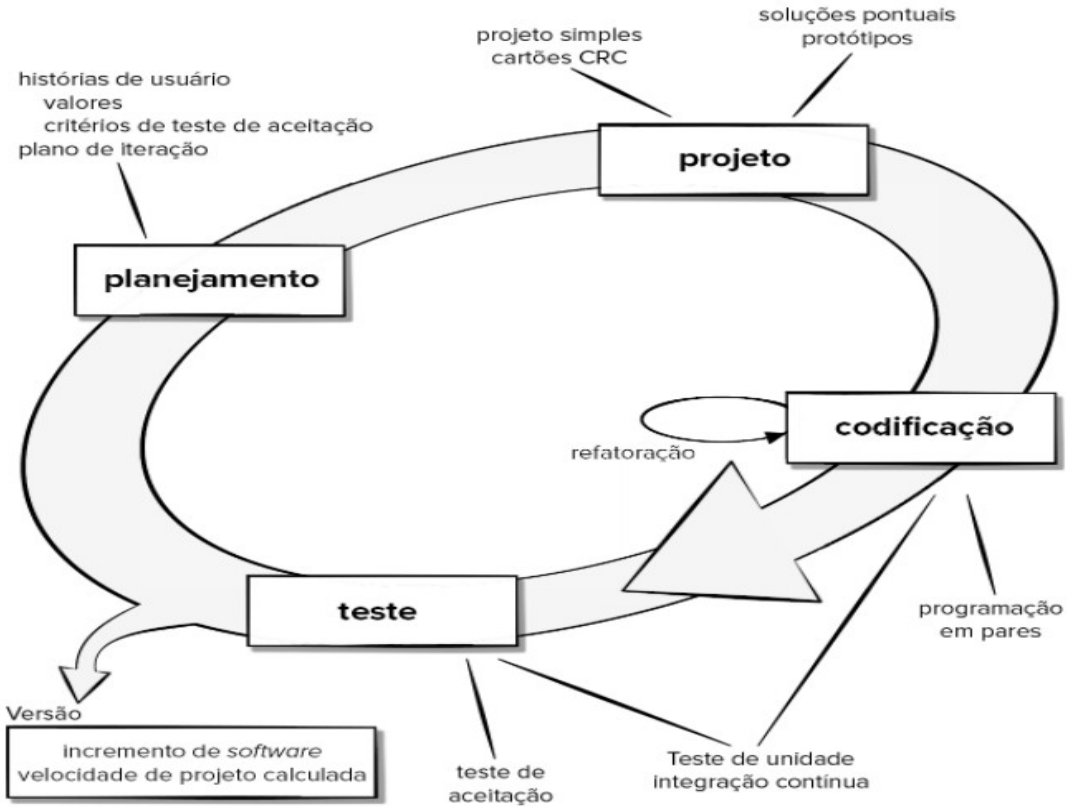


Figura 3.3
O processo da Extreme Programming (XP).

Presman, 2021:46

Extreme Programming (XP)

- Planejamento
  - Os requisitos são gravados em cartões de história e as histórias que serão incluídas em um release são determinadas pelo tempo disponível e sua relativa prioridade.
- Pequenos releases
  - Desenvolve-se um conjunto mínimo de funcionalidades útil, que fornece o valor do negócio. Releases do sistema são frequentes e gradualmente adicionam funcionalidade ao primeiro release

Sommerville, 2011

Extreme Programming (XP)

- Projeto simples
  - Cada projeto é realizado para atender às necessidades atuais, e nada mais.

ID: 1	Classe: Usuários	Super Classe:
Responsabilidades		Colaborações
Conhecer seus dados		
Conhecer se está ativo		
Conhecer seu estado civil		Civil
Conhecer sua renda		Renda
Conhecer sua escoladirade		Escolaridade

Figura 01 – Exemplo cartão CRC(Classe, Responsabilidade e Colaboração)

Extreme Programming (XP)

- Desenvolvimento test-first
  - Um framework de testes iniciais automatizados é usado para escrever os testes para uma nova funcionalidade antes que a funcionalidade em si seja implementada

Sommerville, 2011

Extreme Programming (XP): TDD - Test Driven Development

- Técnica de desenvolvimento de software que baseia em um ciclo curto de repetições:
  - ✓ O desenvolvedor escreve um caso de teste automatizado que define uma melhoria desejada ou uma nova funcionalidade.
  - ✓ É produzido código que possa ser validado pelo teste para posteriormente o código ser refatorado para um código sob padrões aceitáveis.

- Kent Beck, considerado o criador da técnica, declarou em 2003 que TDD encoraja designs de código simples e inspira confiança.

- Através de TDD, programadores podem aplicar o conceito de melhorar e depurar código legado desenvolvido a partir de técnicas antigas.

TDD: Exemplo

Calculadora que calcula as 4 operações: CLASSE DE TESTE

```java
public class CalculadoraTeste {
    @Test
    public void deveriaSomarDoisValoresPassados() throws Exception {
        int valorA = 1;
        int valorB = 2;
        Calculadora calculadora = new Calculadora();
        int soma = calculadora.soma(valorA, valorB);

        assertEquals(3, soma);
    }
}
```

TDD: Exemplo

Calculadora que calcula as 4 operações: CLASSE CALCULADORA

```
1  public class Calculadora {
2
3      public int soma(int valorA, int valorB) {
4          return valorA + valorB;
5      }
6  }
```

Extreme Programming (XP)

- Refatoração
  - Todos os desenvolvedores devem refatorar o código continuamente assim que encontrarem melhorias de código.
- Programação em pares
  - Os desenvolvedores trabalham em pares, verificando o trabalho dos outros e prestando apoio para um bom trabalho sempre

Sommerville, 2011

Extreme Programming (XP)

- Propriedade coletiva
  - Os pares de desenvolvedores trabalham em todas as áreas do sistema, de modo que não se desenvolvam ilhas de expertise.
- Integração contínua
  - Assim que o trabalho em uma tarefa é concluído, ele é integrado ao sistema como um todo. Após essa integração, todos os testes de unidade do sistema devem passar.

Sommerville, 2011

Extreme Programming (XP)

- Ritmo sustentável
  - Grandes quantidades de horas-extra não são consideradas aceitáveis, pois o resultado final, muitas vezes, é a redução da qualidade do código e da produtividade a médio prazo.
- Cliente no local
  - Um representante do usuário final do sistema (o cliente) deve estar disponível todo o tempo à equipe de XP.
    - Em um processo de Extreme Programming, o cliente é um membro da equipe de desenvolvimento e é responsável por levar a ela os requisitos de sistema para implementação.

Sommerville, 2011

Leitura Recomendada - Busca Ativa XX

SCRUM e XP direto das Trincheiras

Material co

http://www.filosofiacienciaarte.org/attachments/article/1050/ScrumeXPDiretodasTrincheiras%20(1).pdf

Leitura Recomendada - Busca Ativa

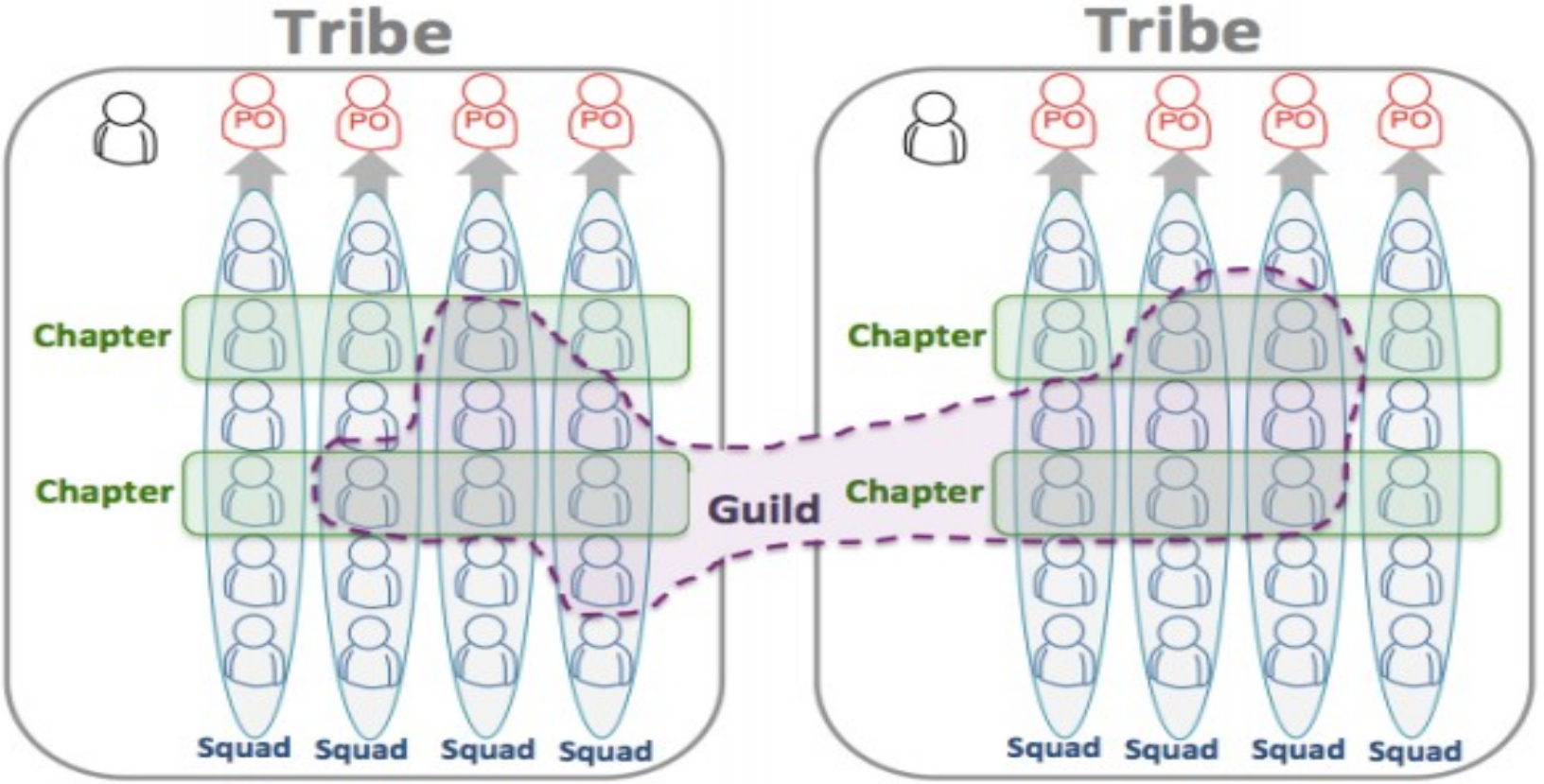
Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software

http://periodicosibepes.org.br/index.php/reinfo/article/view/146

Spotify Engineering Culture

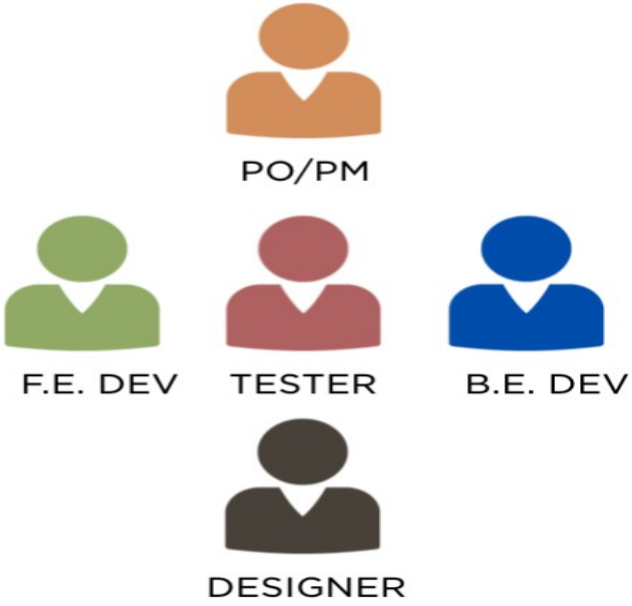
- Também chamado de ***Spotify Scaling Agile*** ou ***Spotify Model***.
- Início em 2012 com a empresa sueca Spotify que seguindo uma cultura ágil de engenharia divulgou como funcionava os seus times.
- Criou o seu próprio processo ágil, criando o ***Squad***, (“Esquadra” ou “Esquadrões”) similar a uma equipe de Scrum.

Spotify Engineering Culture

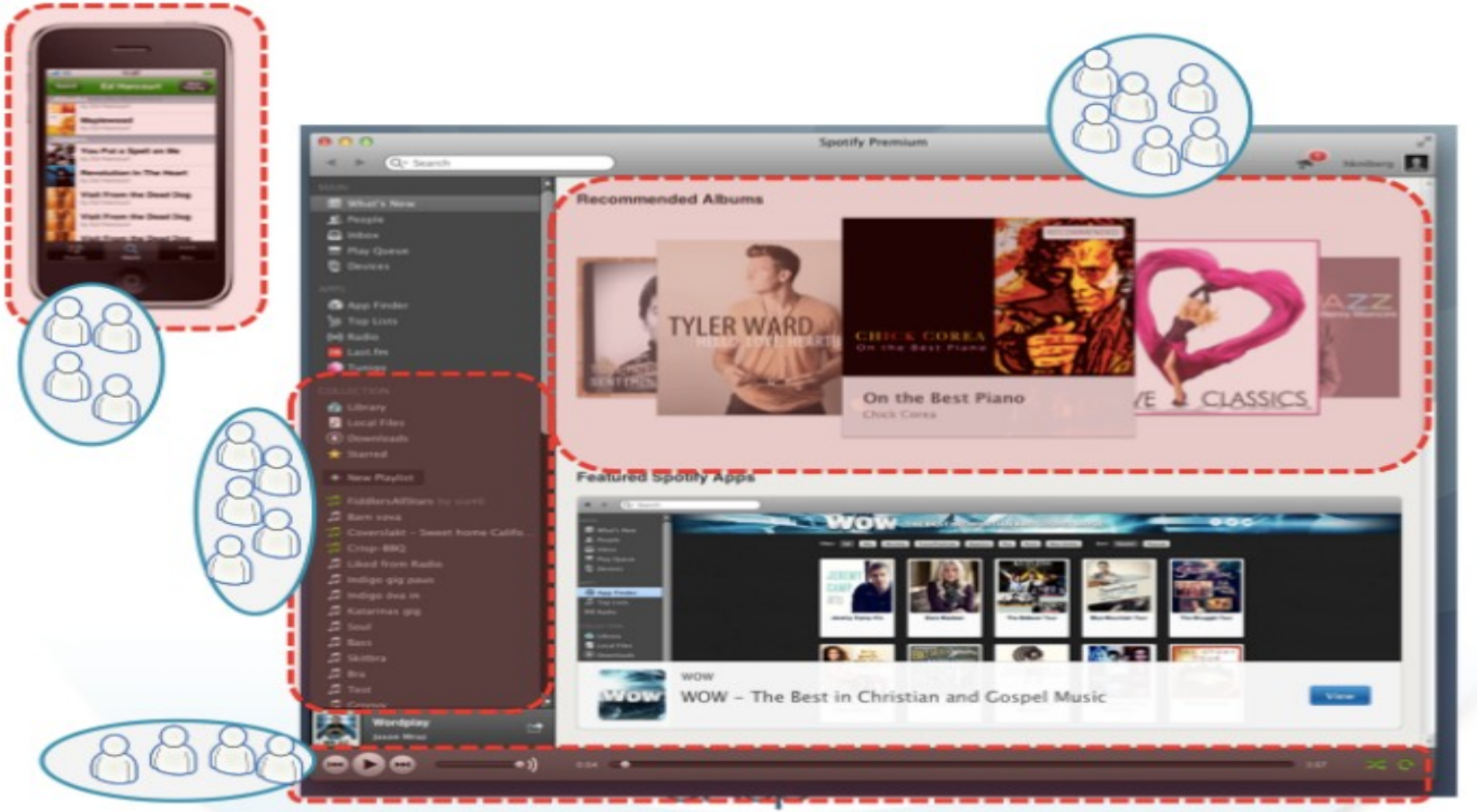


*Spotify Engineering Culture :*Squad

- Equipes **multidisciplinar** independentes e autônomas de 3 a 10 membros para que possam planejar e executar as tarefas com o foco no objetivo final.
- Não tem oficialmente um líder, mas conta com o ***Project Owner (PO)ou Product Management(PM)*** que tem a responsabilidade de organizar as tarefas por ordem de prioridade, porém sem interferir em como elas serão executadas por seus colegas.

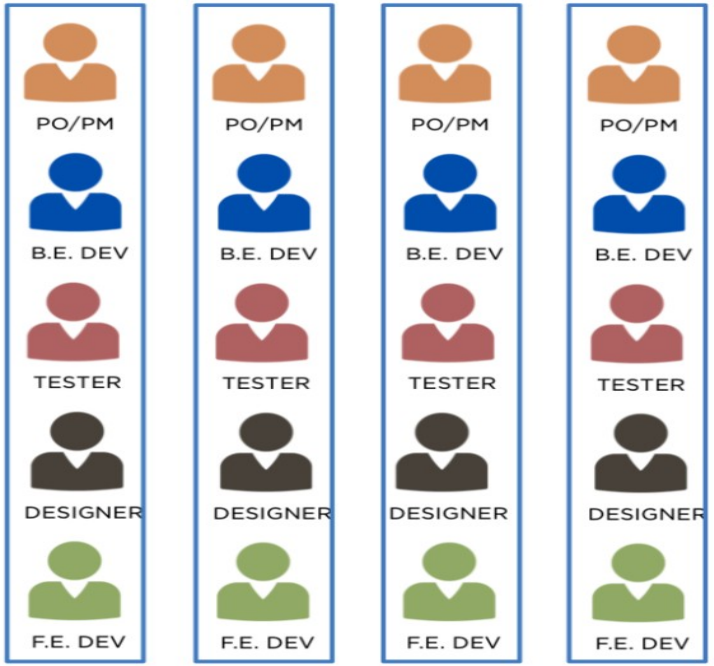


Spotify Engineering Culture: Squad



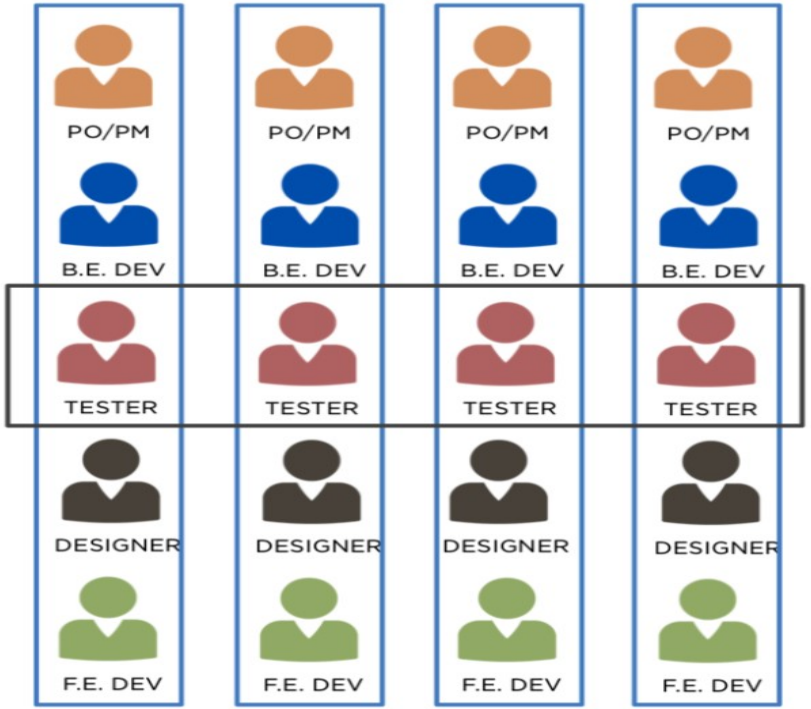
*Spotify Engineering Culture :*Tribes

- Segundo nível de agrupamento, em que pode (e deve) conter uma série de ***squads*** que tenham funcionalidades e objetivos congruentes.
- Um ***Tribe Lead*** irá fornecer orientações e priorização a nível do negócio.
- É indicado que os ***squads*** de uma tribo fiquem próximos fisicamente, para que facilite o acesso e comunicação entre eles.
- Número máximo **100** pessoas.
- **Encontros periódicos** para alinhamento do trabalho, expectativas e troca de experiências.



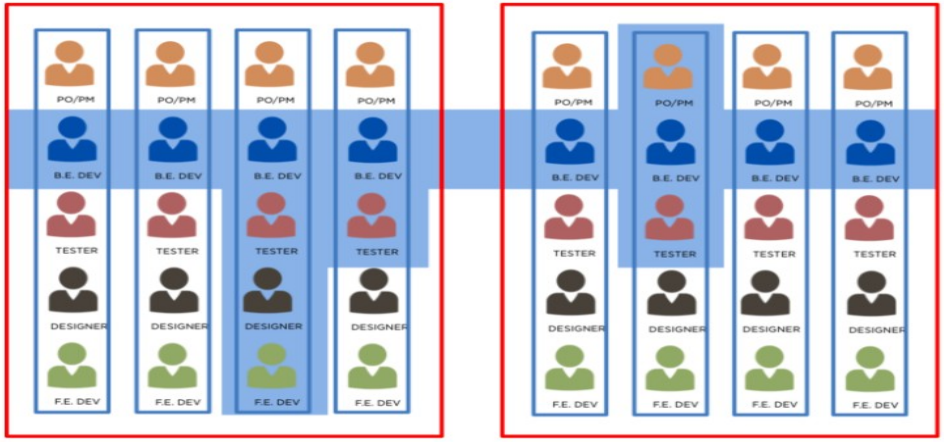
Spotify Engineering Culture: Chapter

- São grupos **transversais**(horizontal) — são definidos por função e skills similares.
  - unem profissionais com as mesmas habilidades.
- É preciso que se encontrem com certa frequência para troca de melhores **práticas** e desafios enfrentados.
- Um líder(***Chapter Leader***), que servirá como ***coaching*** para guiar e orientar os demais membros durante o desenvolvimento do processo.



Spotify Engineering Culture: Guilds

Guildas são os grupos mais difíceis de se definir, pode ser de ***squads*** ou ***tribes*** diferentes, que se reúnem para trocar ideias, experiências, erros, acertos e aprendizados acerca de um tema comum.



Leitura Recomendada - Busca Ativa XX

Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds

https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf

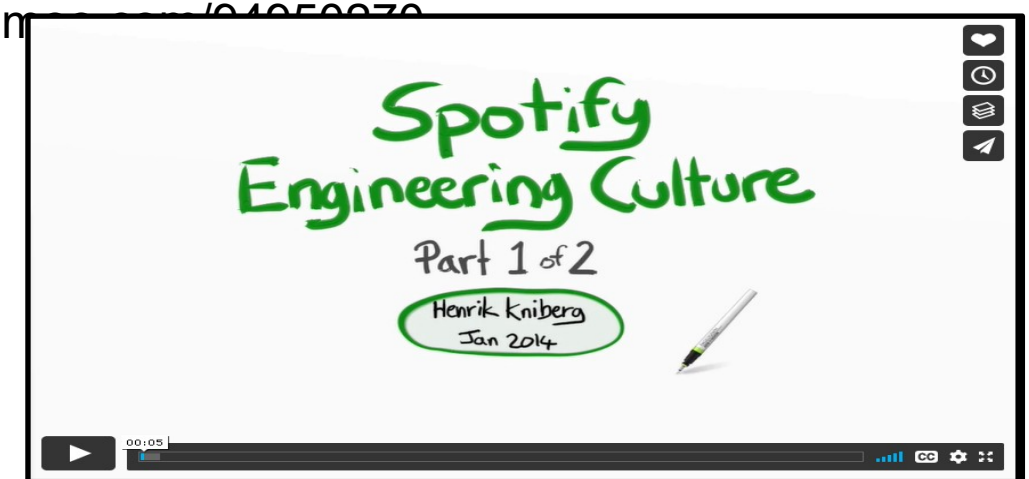
Vídeo Recomendado - Busca Ativa XX

Software Engeneering Culture – parte 1

https://vimeo.com/85490944

Software Engeneering Culture – parte 2

https://vim



Spotify Engineering Culture: Benefícios

- A Spotify foi a pioneira na gestão ágil de projetos utilizando o Squad, o que possibilitou a empresa escalar e buscar a melhoria contínua dos seus produtos.
- Como benefícios ao usar o modelo:
  - Agilidade
  - Motivação
  - Entrega de valor

Spotify Engineering Culture: No Brasil

Várias empresas e startups pelo mundo, inclusive no Brasil, têm adotado o ***Squad***. O importante é testar e criar uma cultura ágil, através de times auto-organizados e com autonomia para conduzir projetos.

- **ContaAzul**, startup de Joinville que lidera uma revolução na gestão financeira de pequenos e médios empresários por todo o Brasil;
- **Nubank**, startup de São Paulo que tem o cartão de crédito mais desejado da história dos serviços bancários.
- **Loft** plataforma digital para simplificar a venda e compra de imóveis
- Mercado Livre, Hostgator

Conclusão

- Métodos ágeis focam nas melhorias e nos requisitos do sistema futuro.
- Deve-se estudar a situação atual sempre que possível.
- O conhecimento adquirido pela revisão do sistema existente pode ser muito valioso para a equipe de projeto.

Dennis, Alan, et al. *Análise e Projeto de Sistemas*. Disponível em: Minha Biblioteca, (5th edição). Grupo GEN, 2014.

Praticando....

- O Cenário apresentado trata da necessidade de um software para controle de uma biblioteca de uma escola pequena que precisa informatizar seus processos de pesquisa ao acervo, realização de empréstimo e devoluções.
- A biblioteca não aplica penalidades aos participantes que atrasam as devoluções.
- Entretanto, mantém em seus arquivos uma restrição vinculada à matrícula do aluno que não devolver o livro até sua formatura e isso implica débitos para o aluno que deseja retirar seu diploma.

Pede-se

- Product backlog (5 funcionalidades)
- Product backlog priorizado
- Estimativa de tarefas
- Sprint Backlog
- Obervações:
  - Política de atendimento 2 minutos por equipe

Referências

- Material de Modelos Métodos e Técnicas de Engenharia de Software cedido pelos professores: Osmar de Oliveira Braz Junior e Richard Henrique de Souza, da Universidade do Sul de Santa Catarina – UNISUL.
- VALENTE, Marco Tulio. Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade, Editora: Independente, 2020
- PRESSMAN, Roger; MAXIM, Bruce. Engenharia de software: uma abordagem profissional. 8.ed. Bookman, 2016. E-book. Disponível em: https://integrada.minhabiblioteca.com.br/books/9788580555349