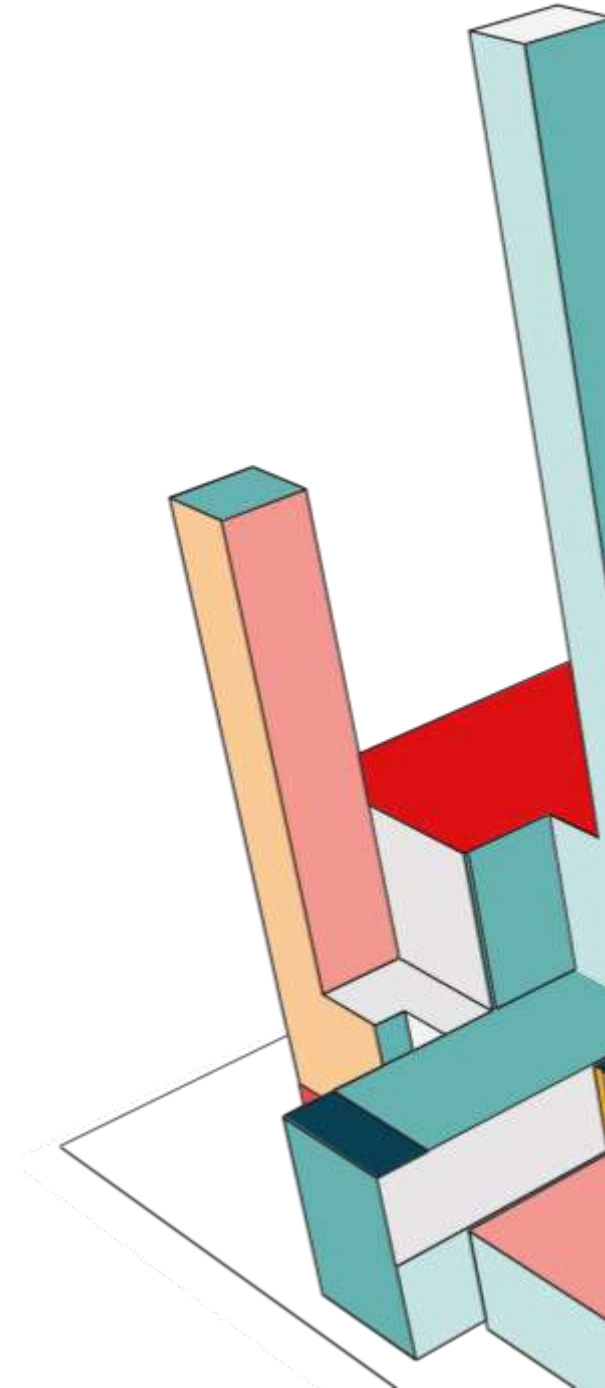


An abstract 3D geometric composition featuring various colored blocks (red, orange, teal, pink, white, and dark blue) arranged in a stepped, architectural style. The blocks are set against a light blue background. The text 'PROCESSO DE DESENVOLVIMENTO DE SOFTWARE' is overlaid on the right side of the image.

# **PROCESSO DE DESENVOLVIMENTO DE SOFTWARE**

# AGENDA

- Introdução
- Características de um engenheiro de software
- A equipe de software
- Estruturas de equipe
- Equipes globais
- Chamada
- Encerramento



# CAPÍTULO 5: ASPECTOS HUMANOS DA ENGENHARIA DE SOFTWARE

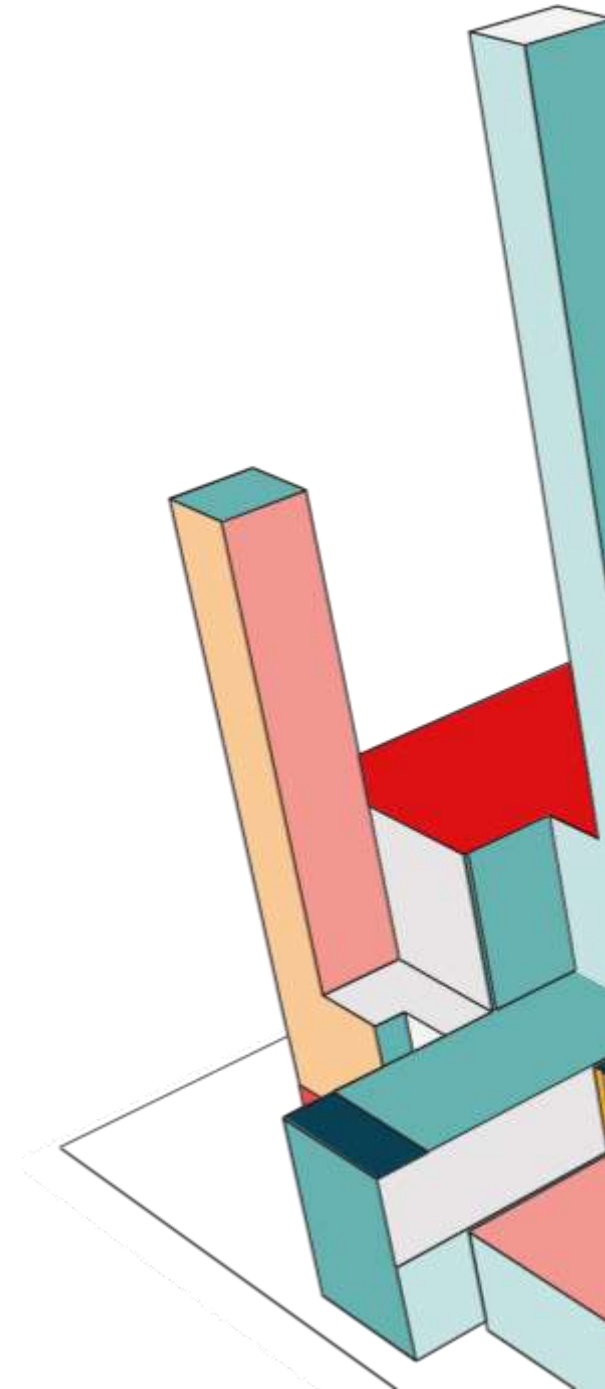




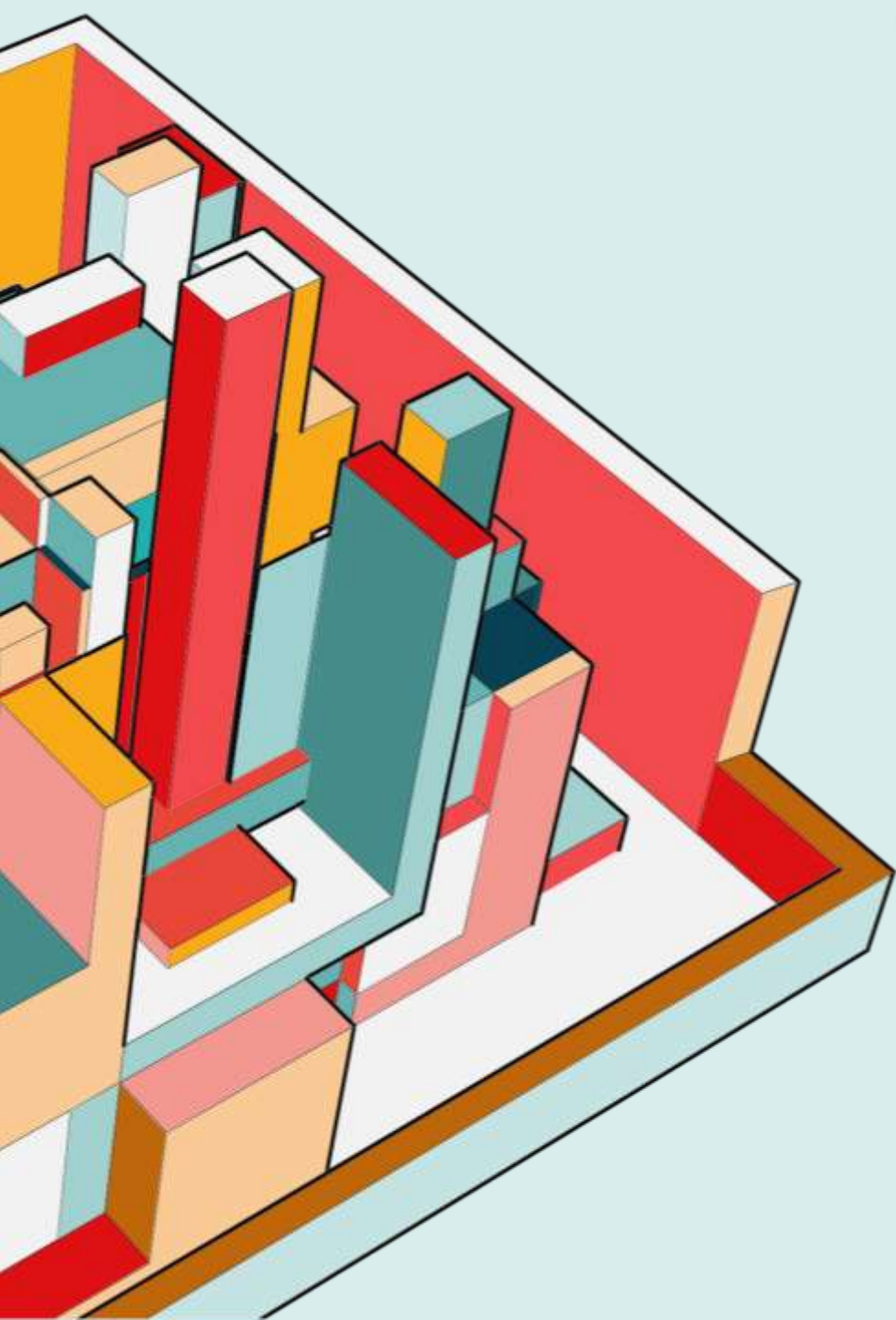
# INTRODUÇÃO

# INTRODUÇÃO

- A engenharia de software tem uma fartura de técnicas, ferramentas e métodos projetados para melhorar tanto o processo de desenvolvimento de software quanto o produto final.
- No entanto, software não é simplesmente um produto de soluções técnicas adequadas aplicadas a hábitos técnicos adequados.
- Software é desenvolvido por pessoas, usado por pessoas e dá suporte à interação entre pessoas.
- Assim, características, comportamento e cooperação humanos são fundamentais no desenvolvimento prático de software.







# CARACTERÍSTICAS DE UM ENGENHEIRO DE SOFTWARE

# ENTÃO VOCÊ QUER SER ENGENHEIRO DE SOFTWARE?

Então você precisa:

- dominar o material técnico
- aprender as habilidades exigidas para entender o problema
- projetar uma solução eficaz
- construir o software e testá-lo com a finalidade de produzir a mais alta qualidade possível

# ENTÃO VOCÊ QUER SER ENGENHEIRO DE SOFTWARE?

Então você precisa:

- dominar o material técnico
- aprender as habilidades exigidas para entender o problema
- projetar uma solução eficaz
- construir o software e testá-lo com a finalidade de produzir a mais alta qualidade possível

Além disso, você precisa:

- gerenciar mudanças
- comunicar-se com os envolvidos
- usar ferramentas adequadas quando necessário



# QUAIS OUTRAS CARACTERÍSTICAS SÃO IMPORTANTES?

Existem outras coisas igualmente importantes – os aspectos humanos que o tornarão um engenheiro de software competente.

Vejamos **sete características pessoais** que estão presentes quando um engenheiro de software demonstra comportamento **profissional e empreendedor**.



# QUAIS OUTRAS CARACTERÍSTICAS SÃO IMPORTANTES?

1. Responsabilidade individual

Um engenheiro de software competente tem um senso de responsabilidade individual.

2. Consciência aguçada

- Isso implica a determinação de cumprir suas promessas para colegas, para os envolvidos e para a gerência.

3. Honesto

4. Resiliência sob pressão

- Significa que ele fará o que precisar ser feito, quando for necessário, executando um esforço adicional para a obtenção de um resultado bem-sucedido.

5. Elevado senso de lealdade

# QUAIS OUTRAS CARACTERÍSTICAS SÃO IMPORTANTES?

1. Responsabilidade individual
2. Consciência aguçada
3. Honesto
4. Resiliência sob pressão
5. Elevado senso de lealdade

Um engenheiro de software competente tem consciência aguçada das necessidades dos outros membros de sua equipe, dos envolvidos que solicitam mudanças a uma solução existente e dos gerentes que têm controle global sobre o projeto.

- Ele observa o ambiente em que as pessoas trabalham e adapta seu comportamento a ele e às próprias pessoas.

# QUAIS OUTRAS CARACTERÍSTICAS SÃO IMPORTANTES?

1. Responsabilidade individual

2. Consciência aguçada

3. Honesto

4. Resiliência sob pressão

5. Elevado senso de lealdade

Um engenheiro de software competente é extremamente honesto.

- Se ele vê um projeto falho, aponta os defeitos de maneira construtiva, mas honesta.
- Se instado a distorcer fatos sobre cronogramas, recursos, desempenho ou outras características do produto ou projeto, opta por ser realista e sincero.

# QUAIS OUTRAS CARACTERÍSTICAS SÃO IMPORTANTES?

1. Responsabilidade individual

2. Consciência aguçada

3. Honesto

4. Resiliência sob pressão

5. Elevado senso de lealdade

Um engenheiro de software competente mostra resiliência sob pressão.

- A engenharia de software está sempre à beira do caos.
- A pressão vem em muitas formas – mudanças nos requisitos e nas prioridades, envolvidos exigentes, um gerente autoritário.
- Um engenheiro de software competente é capaz de suportar a pressão de modo que seu desempenho não seja prejudicado.

# QUAIS OUTRAS CARACTERÍSTICAS SÃO IMPORTANTES?

1. Responsabilidade individual
2. Consciência aguçada
3. Honesto
4. Resiliência sob pressão
5. Elevado senso de lealdade

Um engenheiro de software competente tem elevado senso de lealdade.

- De boa vontade, compartilha os créditos com seus colegas.
- Tenta evitar conflitos de interesse e nunca age no sentido de sabotar o trabalho dos outros.



# QUAIS OUTRAS CARACTERÍSTICAS SÃO IMPORTANTES?

6. Atenção aos detalhes

Um engenheiro de software competente mostra atenção aos detalhes.

7. Pragmático

- Isso não significa obsessão com a perfeição.
- Ele considera atentamente os critérios mais amplos (p. ex., desempenho, custo, qualidade) que foram estabelecidos para o produto e para o projeto nas suas decisões técnicas diárias.

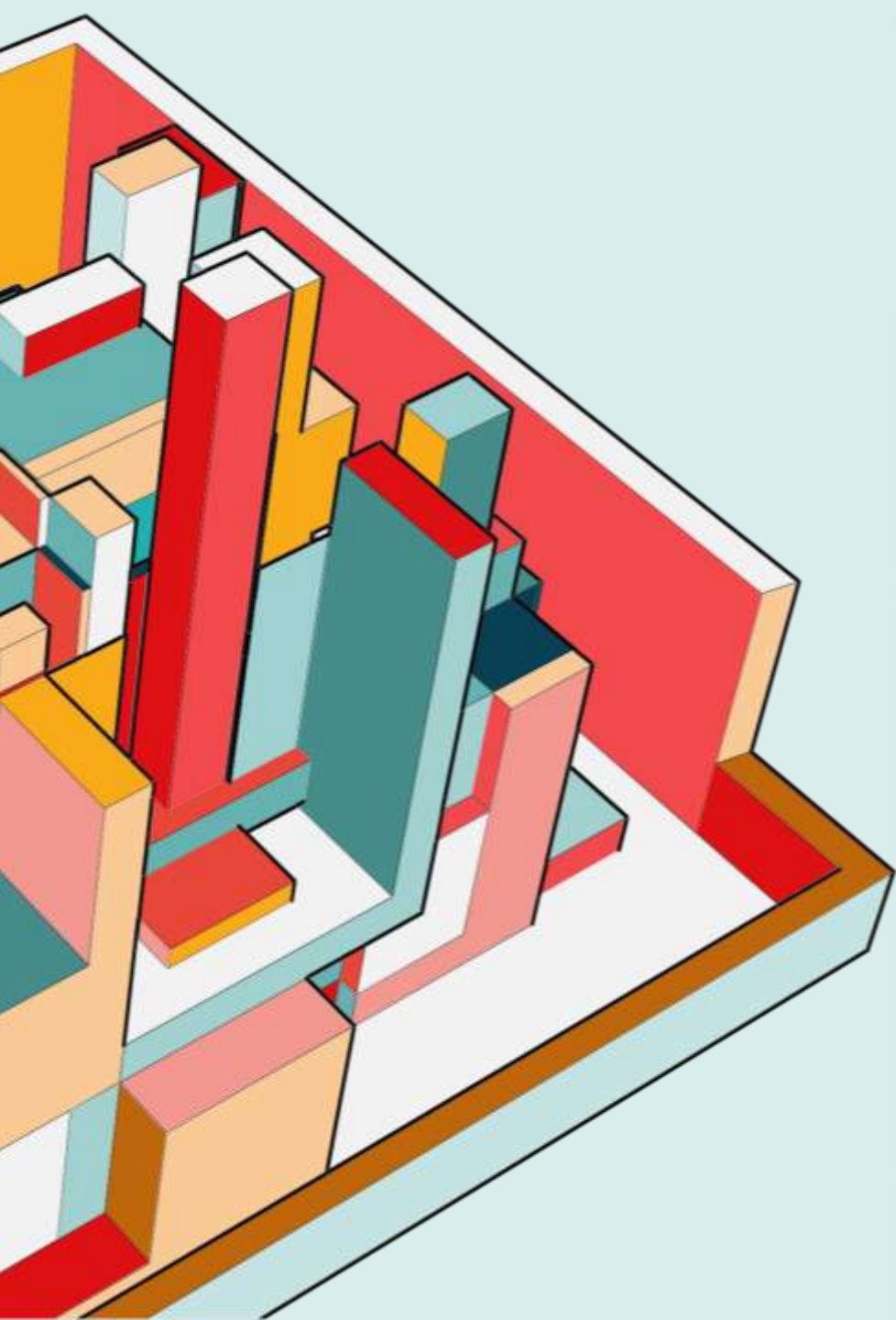
# QUAIS OUTRAS CARACTERÍSTICAS SÃO IMPORTANTES?

6. Atenção aos detalhes

7. Pragmático

Por último, um engenheiro de software competente é pragmático.

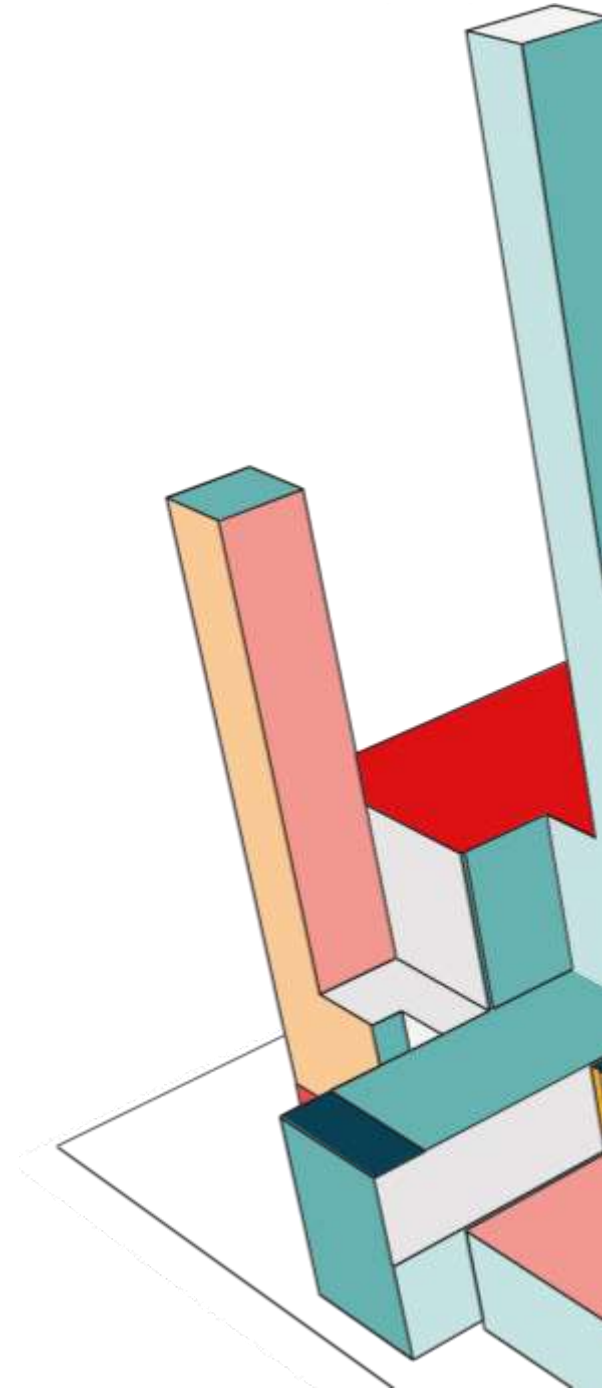
- Reconhece que a engenharia de software não é uma religião na qual devem ser seguidas regras dogmáticas, mas sim uma disciplina que pode ser adaptada de acordo com as circunstâncias.



# A EQUIPE DE SOFTWARE

# A EQUIPE DE SOFTWARE

- Alguns estudos sugerem que os membros de equipes consistentes são significativamente mais produtivos e mais motivados do que a média.
- Compartilham um objetivo comum, uma cultura comum e, em muitos casos, um senso de pertencimento a uma equipe de elite que os torna únicos.



# MAS O QUE TORNA UM EQUIPE DE DESENVOLVIMENTO CONSISTENTE?

- Existem atributos que são normalmente encontrados em equipes de software eficazes.
- Uma equipe de software eficiente deve estabelecer um senso de propósito.

# MAS O QUE TORNA UM EQUIPE DE DESENVOLVIMENTO CONSISTENTE?

- Existem atributos que são normalmente encontrados em equipes de software eficazes.
- Uma equipe de software eficiente deve estabelecer um senso de propósito.
- Uma equipe eficaz também deve ter um senso de envolvimento que permita a cada membro sentir que suas qualidades e contribuições são valiosas.



# MAS O QUE TORNA UM EQUIPE DE DESENVOLVIMENTO CONSISTENTE?

- Existem atributos que são normalmente encontrados em equipes de software eficazes.
- Uma equipe de software eficiente deve estabelecer um senso de propósito.
- Uma equipe eficaz também deve ter um senso de envolvimento que permita a cada membro sentir que suas qualidades e contribuições são valiosas.
- Uma equipe eficaz deve promover um senso de confiança.

# MAS O QUE TORNA UM EQUIPE DE DESENVOLVIMENTO CONSISTENTE?

- Existem atributos que são normalmente encontrados em equipes de software eficazes.
- Uma equipe de software eficiente deve estabelecer um senso de propósito.
- Uma equipe eficaz também deve ter um senso de envolvimento que permita a cada membro sentir que suas qualidades e contribuições são valiosas.
- Uma equipe eficaz deve promover um senso de confiança.
- Os engenheiros de software da equipe devem confiar nas habilidades e na competência de seus colegas e gerentes.

# MAS O QUE TORNA UM EQUIPE DE DESENVOLVIMENTO CONSISTENTE?

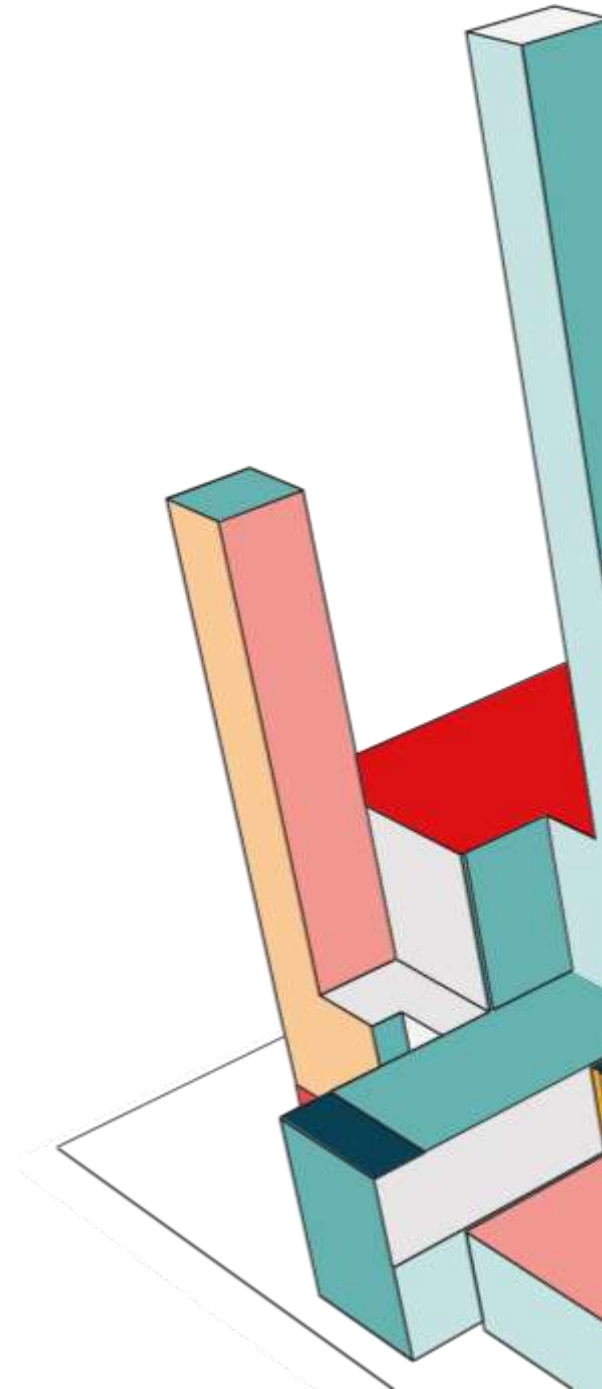
- Existem atributos que são normalmente encontrados em equipes de software eficazes.
- A equipe deve estimular um senso de melhoria, refletindo periodicamente em sua abordagem de engenharia de software e buscando maneiras de melhorar seu trabalho.

# MAS O QUE TORNA UM EQUIPE DE DESENVOLVIMENTO CONSISTENTE?

- Existem atributos que são normalmente encontrados em equipes de software eficazes.
- As equipes de software mais eficazes são diversificadas, no sentido de combinarem uma variedade de diferentes qualidades.
  - Técnicos altamente capacitados são complementados por membros que podem ter menos base técnica, mas compreendem melhor as necessidades dos envolvidos.

# E AS EQUIPES INCONSISTENTES, EXISTEM?

- Sim, nem todas as equipes são eficazes e nem todas são consistentes.
- Na verdade, muitas sofrem de **toxicidade de equipe**.



# E AS EQUIPES INCONSISTENTES, EXISTEM?

Os cinco fatores que promovem um ambiente em equipe potencialmente tóxico são:

1. uma atmosfera de trabalho frenética



# E AS EQUIPES INCONSISTENTES, EXISTEM?

Os cinco fatores que promovem um ambiente em equipe potencialmente tóxico são:

1. uma atmosfera de trabalho frenética
2. alto grau de frustração que causa atrito entre os membros da equipe

# E AS EQUIPES INCONSISTENTES, EXISTEM?

Os cinco fatores que promovem um ambiente em equipe potencialmente tóxico são:

1. uma atmosfera de trabalho frenética
2. alto grau de frustração que causa atrito entre os membros da equipe
3. um processo de software fragmentado ou coordenado de forma deficiente

# E AS EQUIPES INCONSISTENTES, EXISTEM?

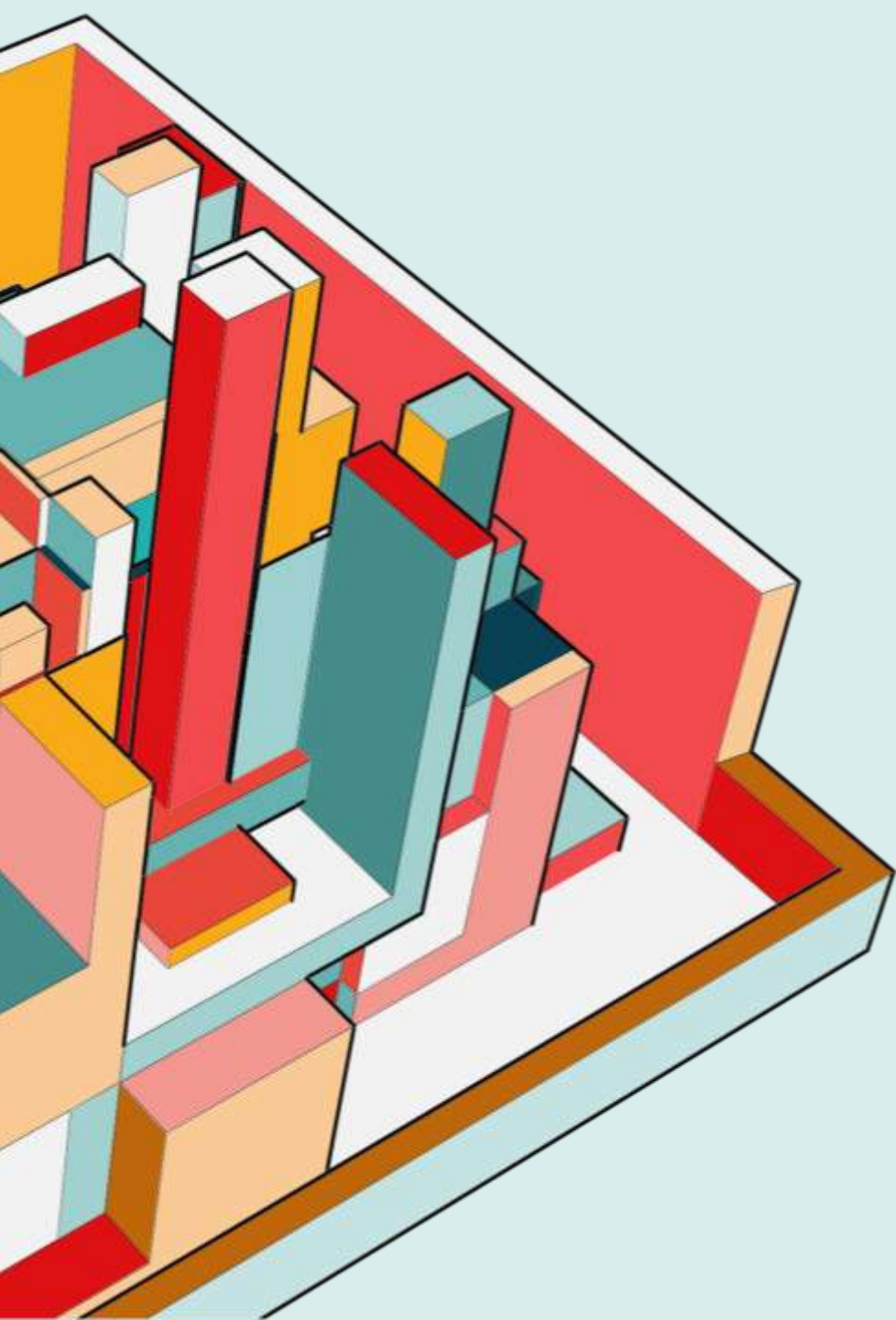
Os cinco fatores que promovem um ambiente em equipe potencialmente tóxico são:

1. uma atmosfera de trabalho frenética
2. alto grau de frustração que causa atrito entre os membros da equipe
3. um processo de software fragmentado ou coordenado de forma deficiente
4. uma definição nebulosa dos papéis dentro da equipe de software

# E AS EQUIPES INCONSISTENTES, EXISTEM?

Os cinco fatores que promovem um ambiente em equipe potencialmente tóxico são:

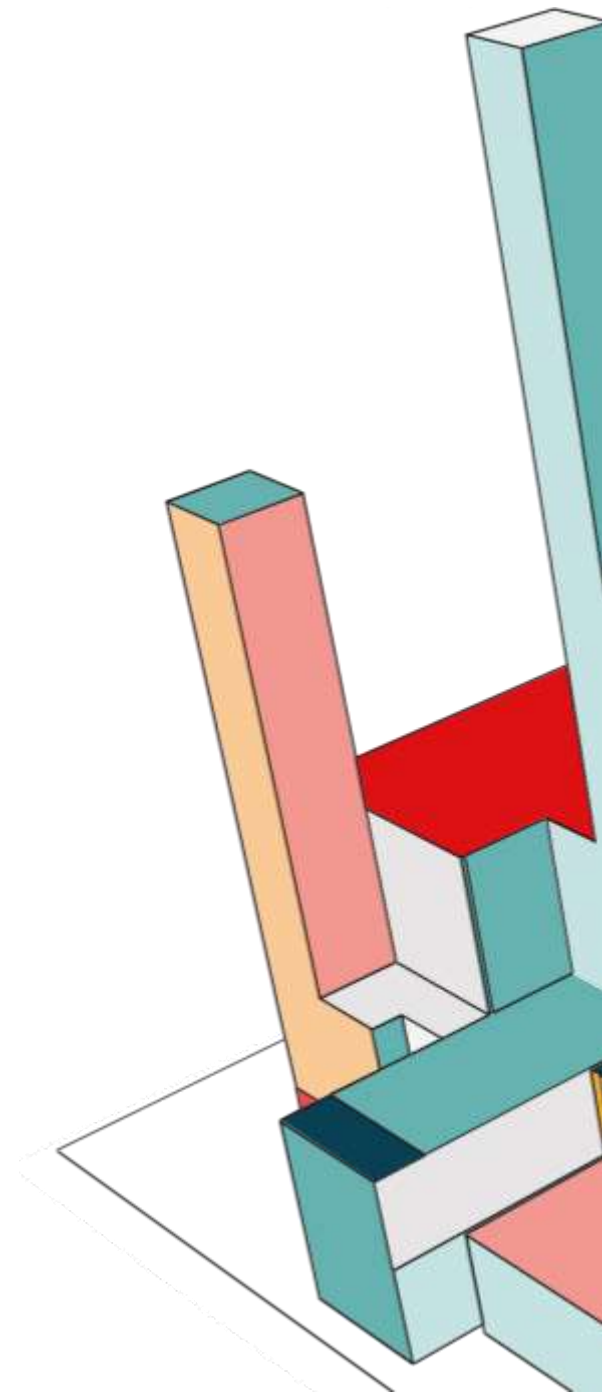
1. uma atmosfera de trabalho frenética
2. alto grau de frustração que causa atrito entre os membros da equipe
3. um processo de software fragmentado ou coordenado de forma deficiente
4. uma definição nebulosa dos papéis dentro da equipe de software
5. contínua e repetida exposição a falhas



# **ESTRUTURAS DE EQUIPE**

# O QUE CONSIDERAR AO PLANEJAR A ESTRUTURA DA EQUIPE?

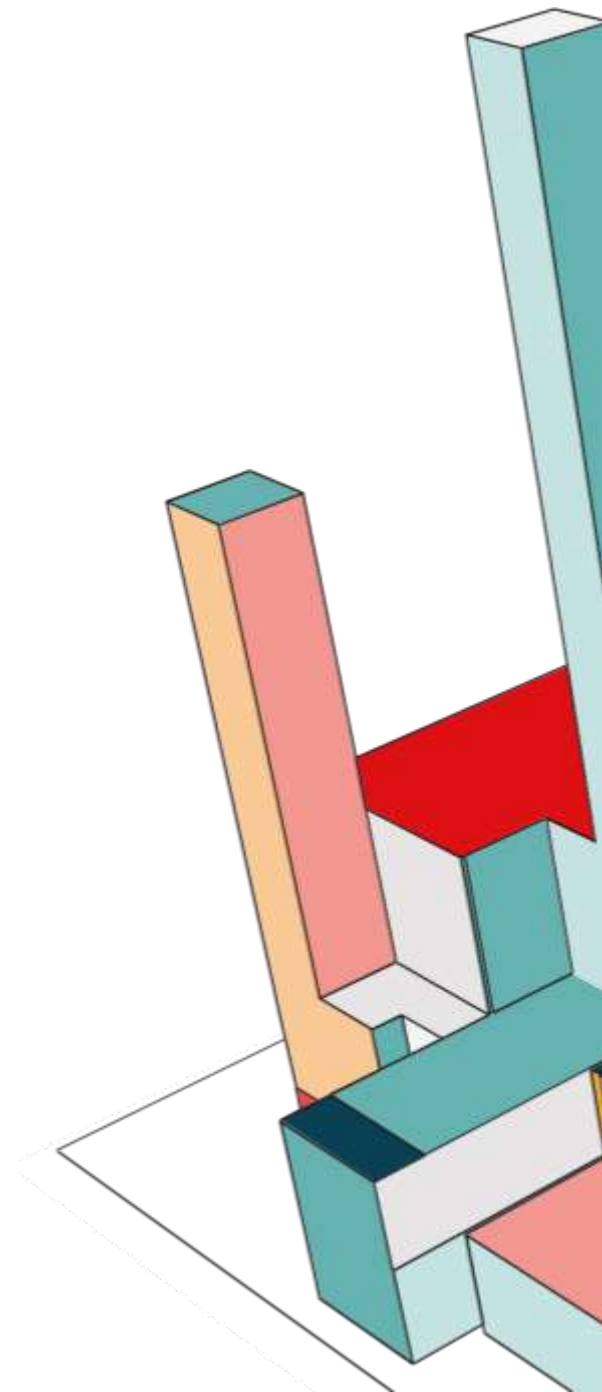
- Vários fatores devem ser considerados ao planejarmos a estrutura da equipe de engenharia de software:
  - dificuldade do problema a ser resolvido
  - “tamanho” do programa (ou programas) resultante em linhas de código ou pontos de função
  - tempo que a equipe irá permanecer reunida (tempo de vida da equipe)
  - até que ponto o problema pode ser modularizado





# O QUE CONSIDERAR AO PLANEJAR A ESTRUTURA DA EQUIPE?

- Vários fatores devem ser considerados ao planejarmos a estrutura da equipe de engenharia de software:
  - qualidade e confiabilidade exigidas do sistema a ser construído
  - rigidez da data de entrega
  - e grau de sociabilidade (comunicação) exigida para o projeto



# EQUIPES ÁGEIS SÃO A SOLUÇÃO?

Ao longo da última década, o desenvolvimento de software ágil tem sido indicado como o antídoto para muitos problemas que se alastraram nas atividades de projeto de software.

# EQUIPES ÁGEIS SÃO A SOLUÇÃO?

Ao longo da última década, o desenvolvimento de software ágil tem sido indicado como o antídoto para muitos problemas que se alastraram nas atividades de projeto de software.

- a satisfação do cliente e a entrega prévia incremental de software

A filosofia ágil enfatiza:

# EQUIPES ÁGEIS SÃO A SOLUÇÃO?

Ao longo da última década, o desenvolvimento de software ágil tem sido indicado como o antídoto para muitos problemas que se alastraram nas atividades de projeto de software.

- a satisfação do cliente e a entrega prévia incremental de software
- pequenas equipes de projeto altamente motivadas

A filosofia ágil enfatiza:

# EQUIPES ÁGEIS SÃO A SOLUÇÃO?

Ao longo da última década, o desenvolvimento de software ágil tem sido indicado como o antídoto para muitos problemas que se alastraram nas atividades de projeto de software.

A filosofia ágil enfatiza:

- a satisfação do cliente e a entrega prévia incremental de software
- pequenas equipes de projeto altamente motivadas
- métodos informais

# EQUIPES ÁGEIS SÃO A SOLUÇÃO?

Ao longo da última década, o desenvolvimento de software ágil tem sido indicado como o antídoto para muitos problemas que se alastraram nas atividades de projeto de software.

A filosofia ágil enfatiza:

- a satisfação do cliente e a entrega prévia incremental de software
- pequenas equipes de projeto altamente motivadas
- métodos informais
- mínimos artefatos de engenharia de software

# EQUIPES ÁGEIS SÃO A SOLUÇÃO?

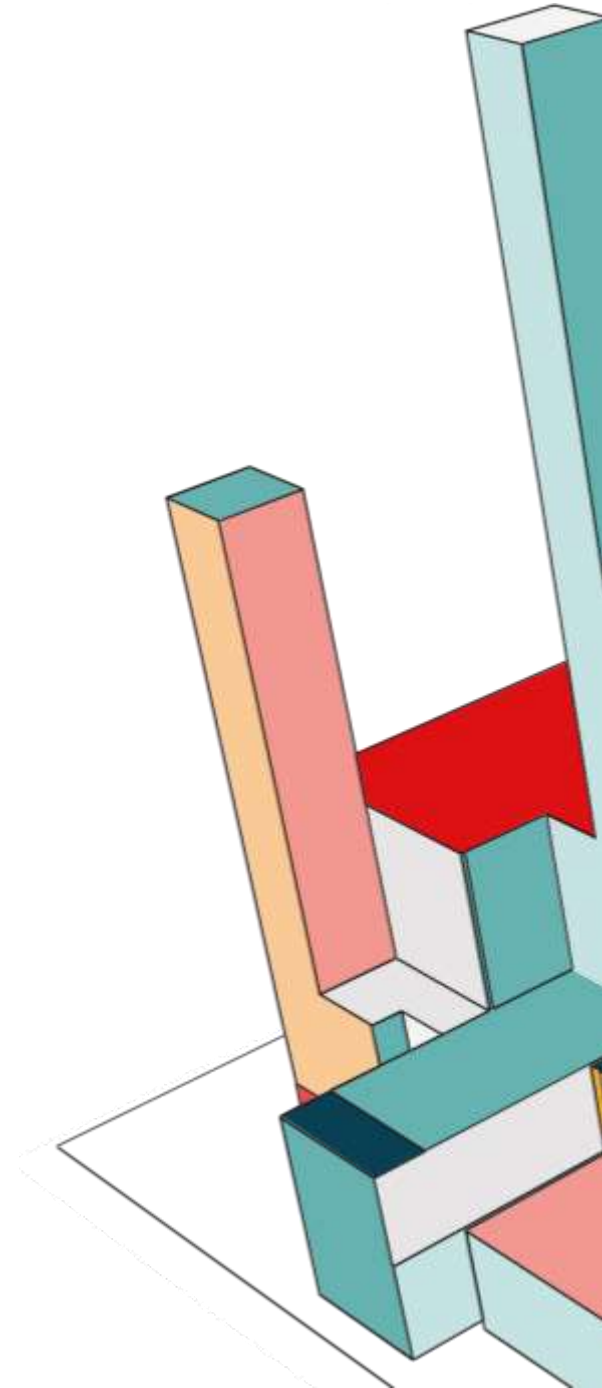
Ao longo da última década, o desenvolvimento de software ágil tem sido indicado como o antídoto para muitos problemas que se alastraram nas atividades de projeto de software.

A filosofia ágil enfatiza:

- a satisfação do cliente e a entrega prévia incremental de software
- pequenas equipes de projeto altamente motivadas
- métodos informais
- mínimos artefatos de engenharia de software
- total simplicidade de desenvolvimento

# EQUIPES ÁGEIS SÃO A SOLUÇÃO?

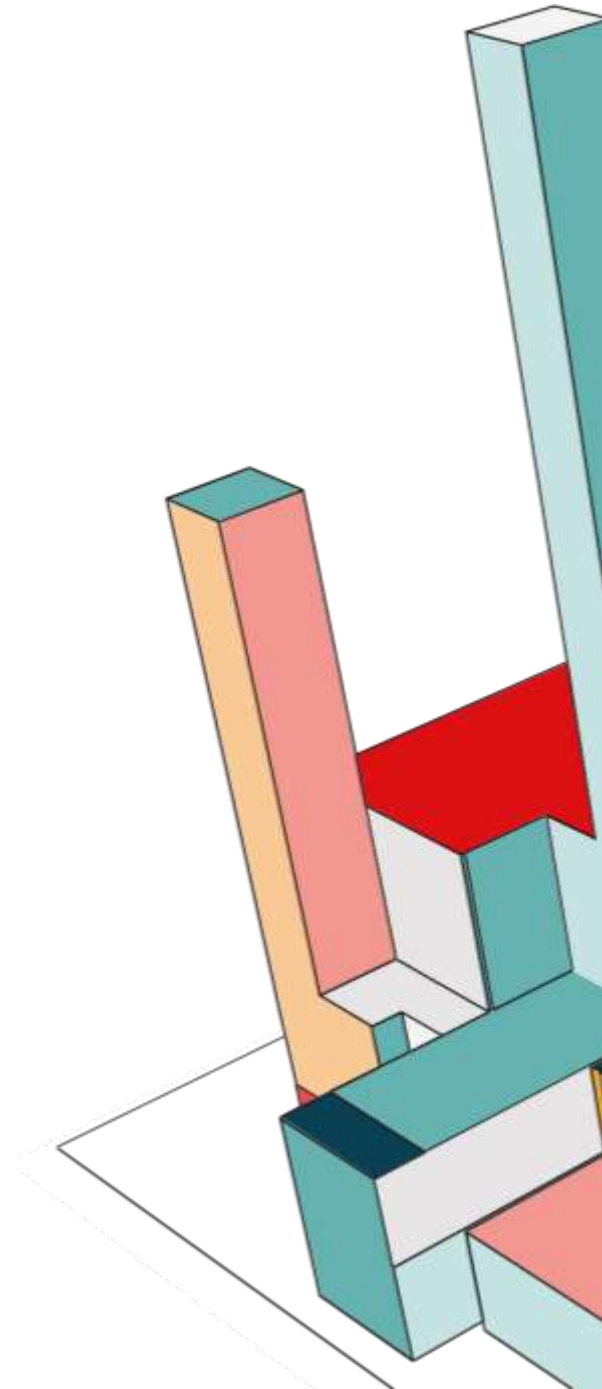
- Para uso das competências de cada membro da equipe e para fomentar colaboração efetiva ao longo do projeto, equipes ágeis são auto-organizadas.





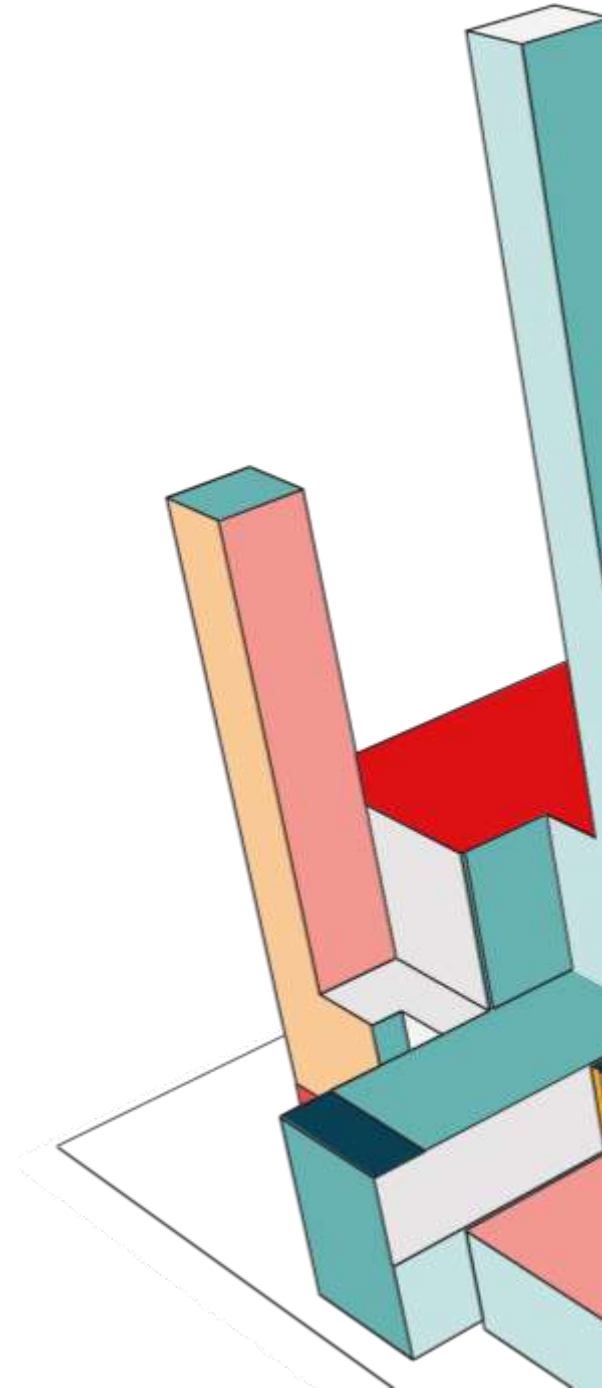
# EQUIPES ÁGEIS SÃO A SOLUÇÃO?

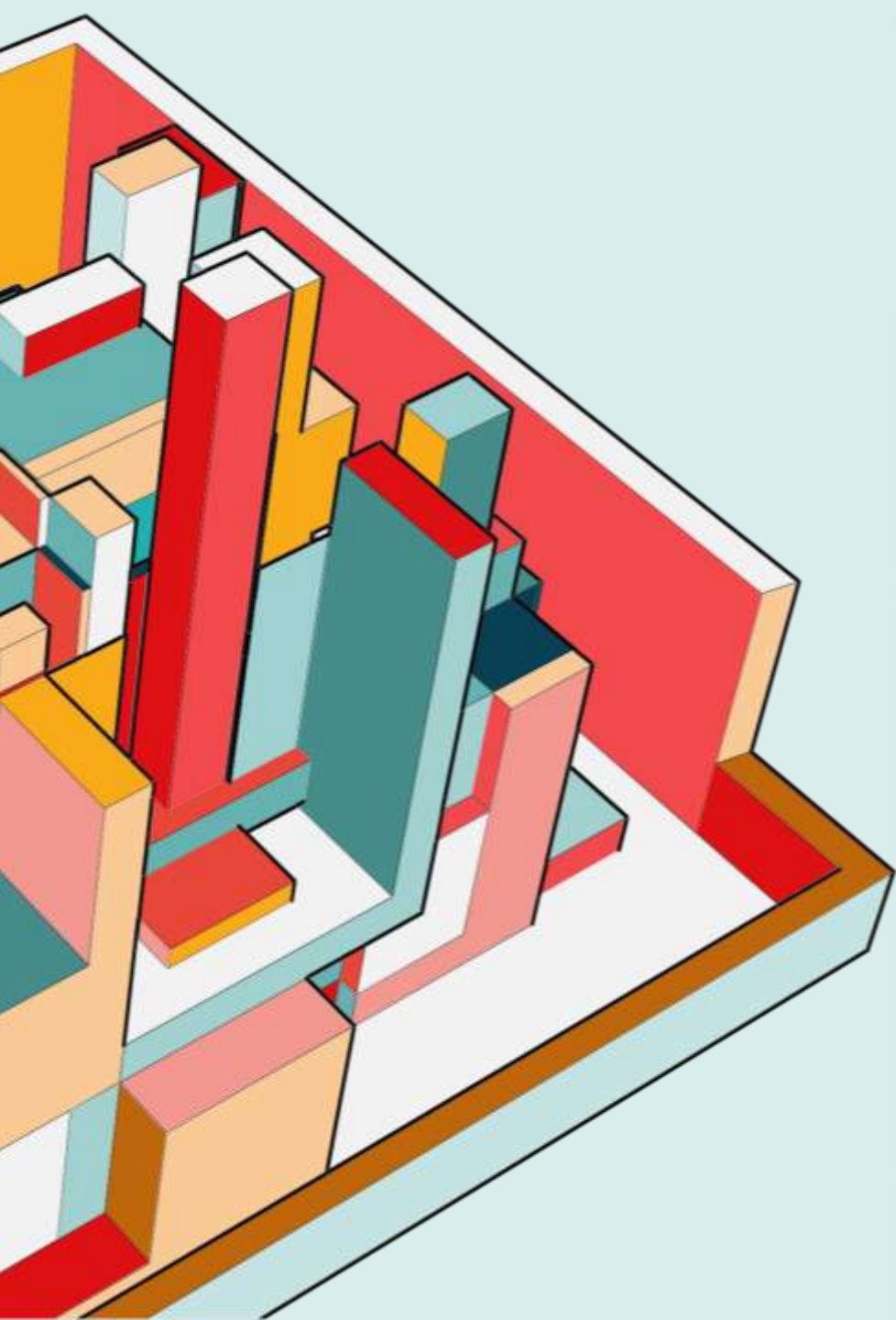
- Para uso das competências de cada membro da equipe e para fomentar colaboração efetiva ao longo do projeto, equipes ágeis são auto-organizadas.
- Uma equipe auto-organizada não mantém, necessariamente, uma estrutura de equipe única.



# EQUIPES ÁGEIS SÃO A SOLUÇÃO?

- Para uso das competências de cada membro da equipe e para fomentar colaboração efetiva ao longo do projeto, equipes ágeis são auto-organizadas.
- Uma equipe auto-organizada não mantém, necessariamente, uma estrutura de equipe única.
- A equipe realiza as alterações necessárias à sua estrutura para reagir às mudanças no ambiente de desenvolvimento ou às mudanças na solução do problema de engenharia em evolução.

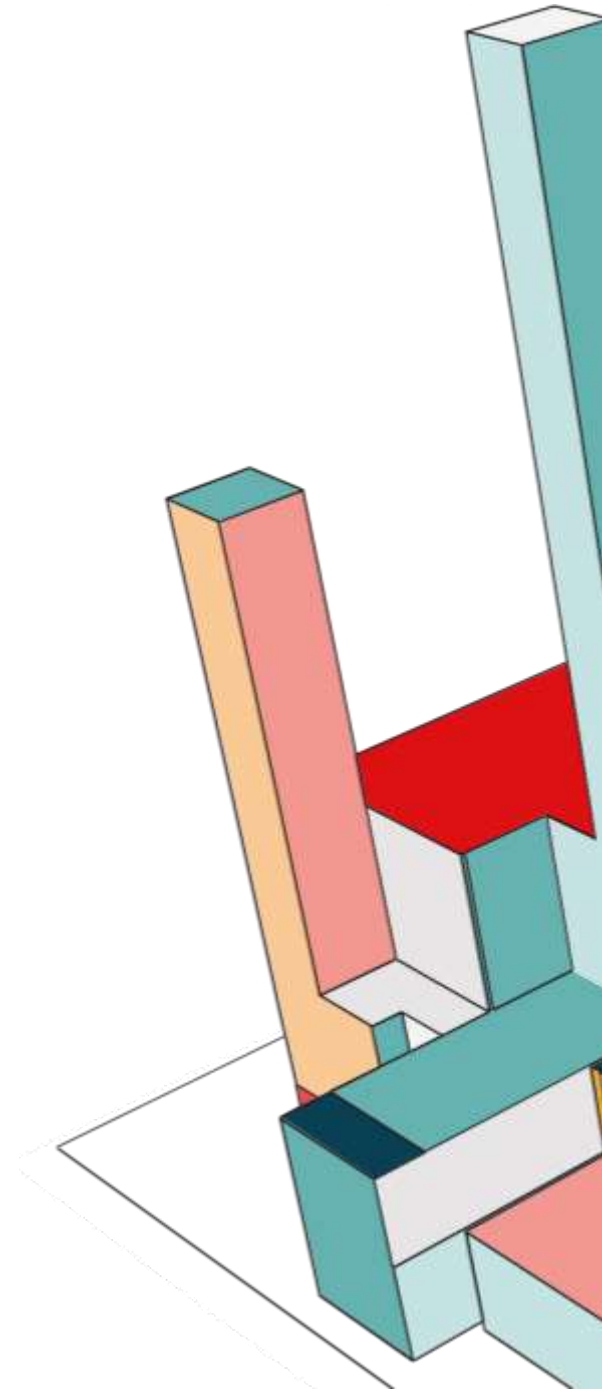




**EQUIPES GLOBAIS**

# EQUIPES GLOBAIS

- No campo do software, globalização significa mais do que a transferência de bens e serviços entre fronteiras internacionais.
- Nas últimas décadas, foi construído um crescente número de importantes produtos de software por equipes muitas vezes sediadas em diferentes países.



# DESAFIOS DA EQUIPE GLOBAL

Essas equipes de desenvolvimento de software global têm desafios exclusivos, que incluem:

- Coordenação
- Colaboração
- Comunicação
- Tomada de decisão especializada

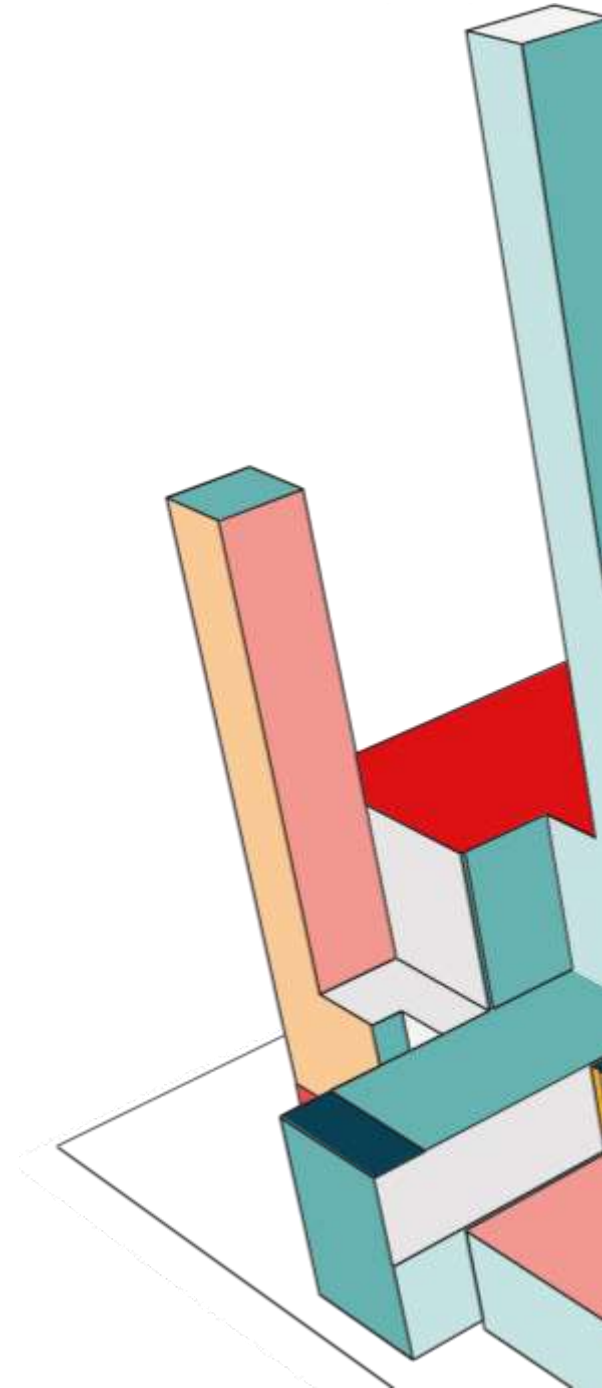
# COMPLICAÇÕES DA TOMADA DE DECISÕES

1. Além disso, a tomada de decisão em todas as equipes de software é complicada por quatro fatores:

1. Complexidade do problema.
2. Incerteza e risco associados à decisão.
3. A lei das consequências não intencionais.
4. Diferentes visões do problema que levam a diferentes conclusões sobre o caminho a seguir.

# CONSEQUÊNCIAS

- Os problemas inerentes a essa dinâmica podem resultar em um projeto instável.



**CHAMADA**





**ENCERRAMENTO**

