

Você já trabalhou, trabalha ou ainda vai
trabalhar em um projeto onde...

O código-fonte é tão bagunçado que você
não sabe nem por onde começar...

Existem partes do código-fonte que só uma
pessoa sabe mexer...

Você tem medo de mexer numa coisa
e estragar outra...

Tem mais **defeitos para corrigir** do que
funcionalidades para implementar...

É necessário fazer horas extras
para conseguir entregar...

Ninguém quer fazer deploy na sexta-feira...

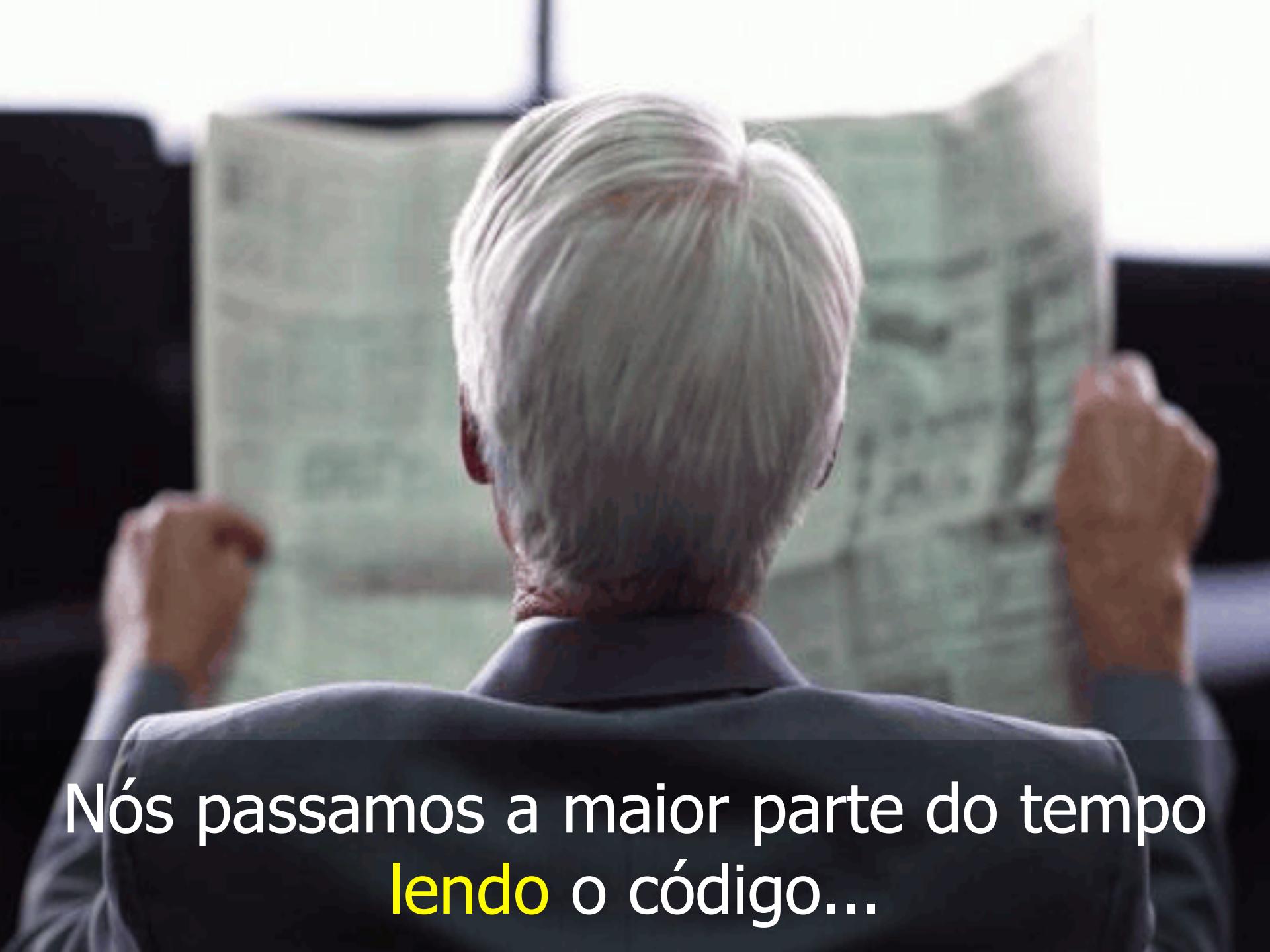


Porque isso acontece?



Podem existir muitos motivos mas sem
dúvida tudo isso tem algo em comum...

```
2445     if ((lowerChar == Character.ERROR) ||
2446         (lowerChar >= Character.MIN_SUPPLEMENTARY_CODE_POINT)) {
2447         if (lowerChar == Character.ERROR) {
2448             lowerCharArray =
2449                 ConditionalSpecialCasing.toLowerCaseCharArray(this, i, locale);
2450         } else if (srcCount == 2) {
2451             resultOffset += Character.toChars(lowerChar, result, i + resultOffset) - srcCount;
2452             continue;
2453         } else {
2454             lowerCharArray = Character.toChars(lowerChar);
2455         }
2456
2457         /* Grow result if needed */
2458         int mapLen = lowerCharArray.length;
2459         if (mapLen > srcCount) {
2460             char[] result2 = new char[result.length + mapLen - srcCount];
2461             System.arraycopy(result, 0, result2, 0,
2462                             i + resultOffset);
2463             result = result2;
2464         }
2465         for (int x=0; x<mapLen; ++x) {
2466             result[i+resultOffset+x] = lowerCharArray[x];
2467         }
2468         resultOffset += (mapLen - srcCount);
2469     } else {
2470         result[i+resultOffset] = (char)lowerChar;
2471     }
2472 }
2473 return new String(0, count+resultOffset, result);
2474 }
```

A photograph showing the back of a person's head and shoulders. The person is looking down at a large, open document or blueprint spread out on a table in front of them. The document appears to be a technical drawing or map, with various lines, shapes, and text visible. The person is wearing a dark-colored shirt.

Nós passamos a maior parte do tempo
lendo o código...



Quem já virou a noite por causa de um **defeito**
que precisava ser corrigido com urgência?



Por esse motivos, o **pair programming**
pode ser uma excelente ferramenta

Vantagens do pair programming:

- Menos interrupções

Vantagens do pair programming:

- Menos interrupções
- Maior qualidade

Vantagens do pair programming:

- Menos interrupções
- Maior qualidade
- Compartilhamento de conhecimento

Vantagens do pair programming:

- Menos interrupções
- Maior qualidade
- Compartilhamento de conhecimento
- Nivelamento técnico



Sensação de estar realizando um trabalho
braçal e desgastante



Maior incidência de **defeitos**

"The logic should be straightforward and make it
hard for bugs to hide"

Bjarne Stroustrup (Criador do C++)



Menos produtividade



Ninguém se sente bem fazendo **gambiarra**



Você já **pediu demissão** alguma vez porque
não aguentava mais o projeto?



Sem dúvida, a qualidade do código-fonte afeta diretamente a **rotatividade** na equipe



O que **motiva** uma pessoa?

O que motiva uma pessoa?

- Dinheiro

O que motiva uma pessoa?

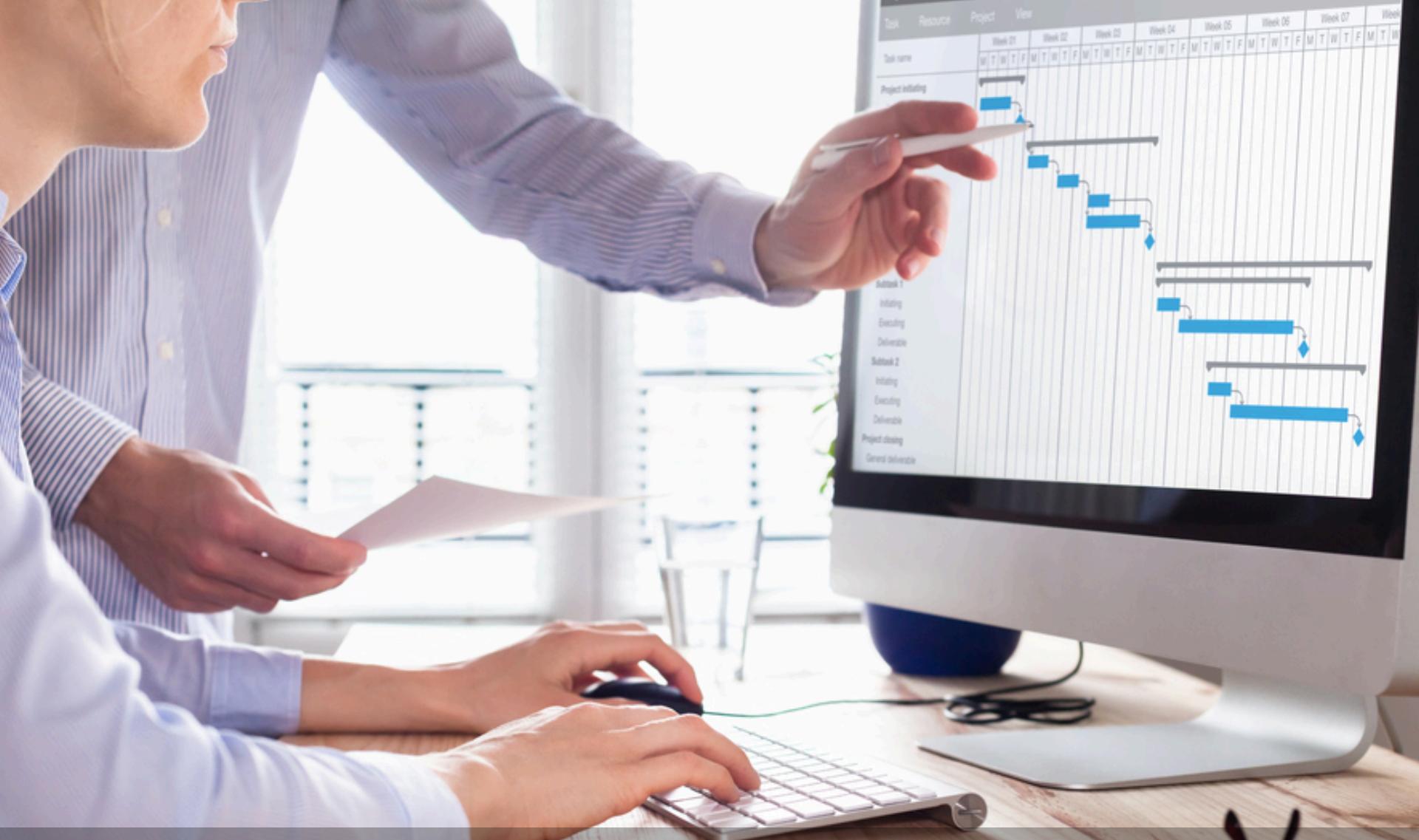
- Dinheiro
- Ambiente de trabalho

O que motiva uma pessoa?

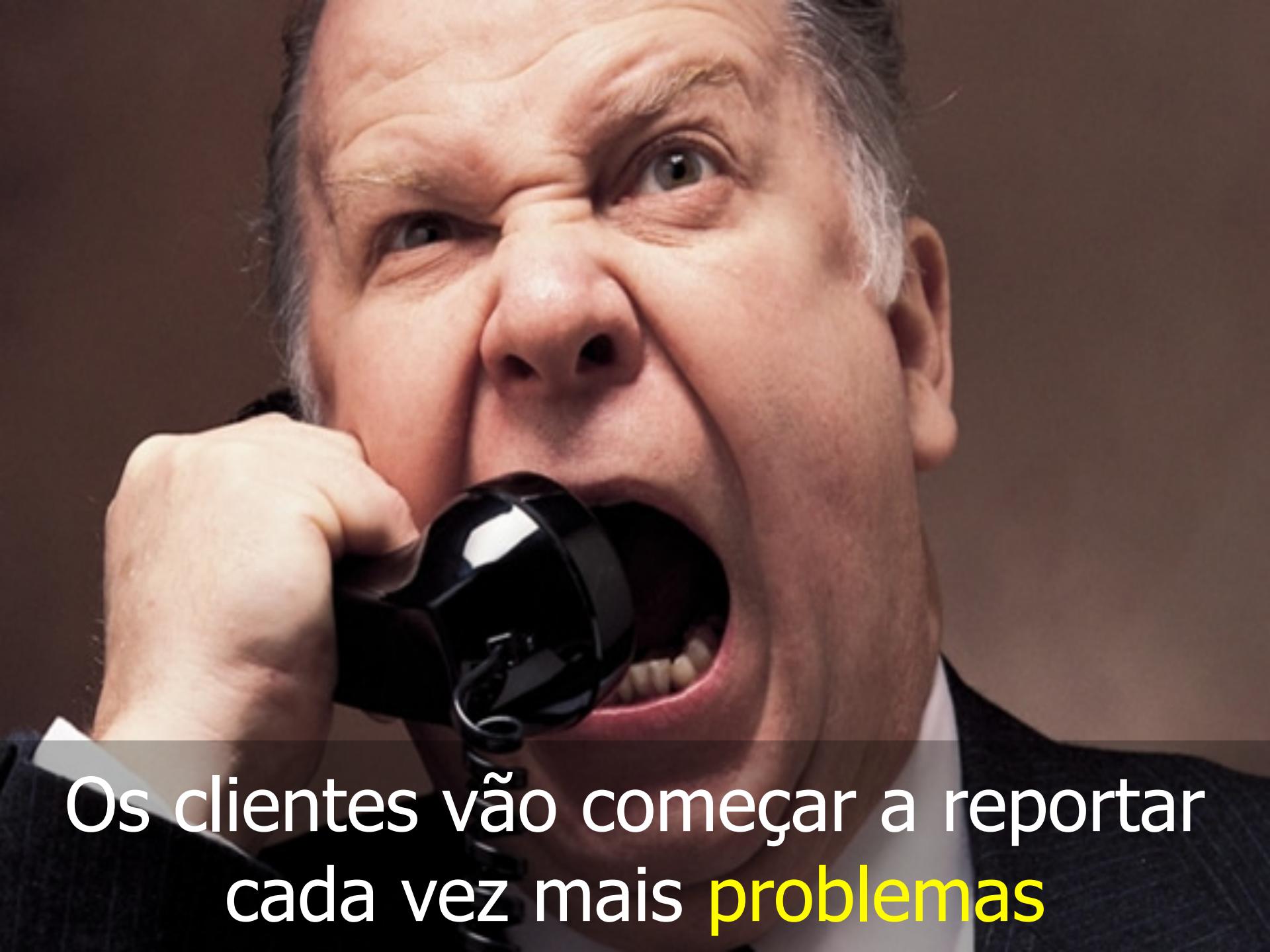
- Dinheiro
- Ambiente de trabalho
- Crescimento profissional



Esse tipo de problema afeta toda a
empresa, não só a área de tecnologia



Como funcionam as **estimativas** na empresa
em que você trabalha?



Os clientes vão começar a reportar
cada vez mais **problemas**



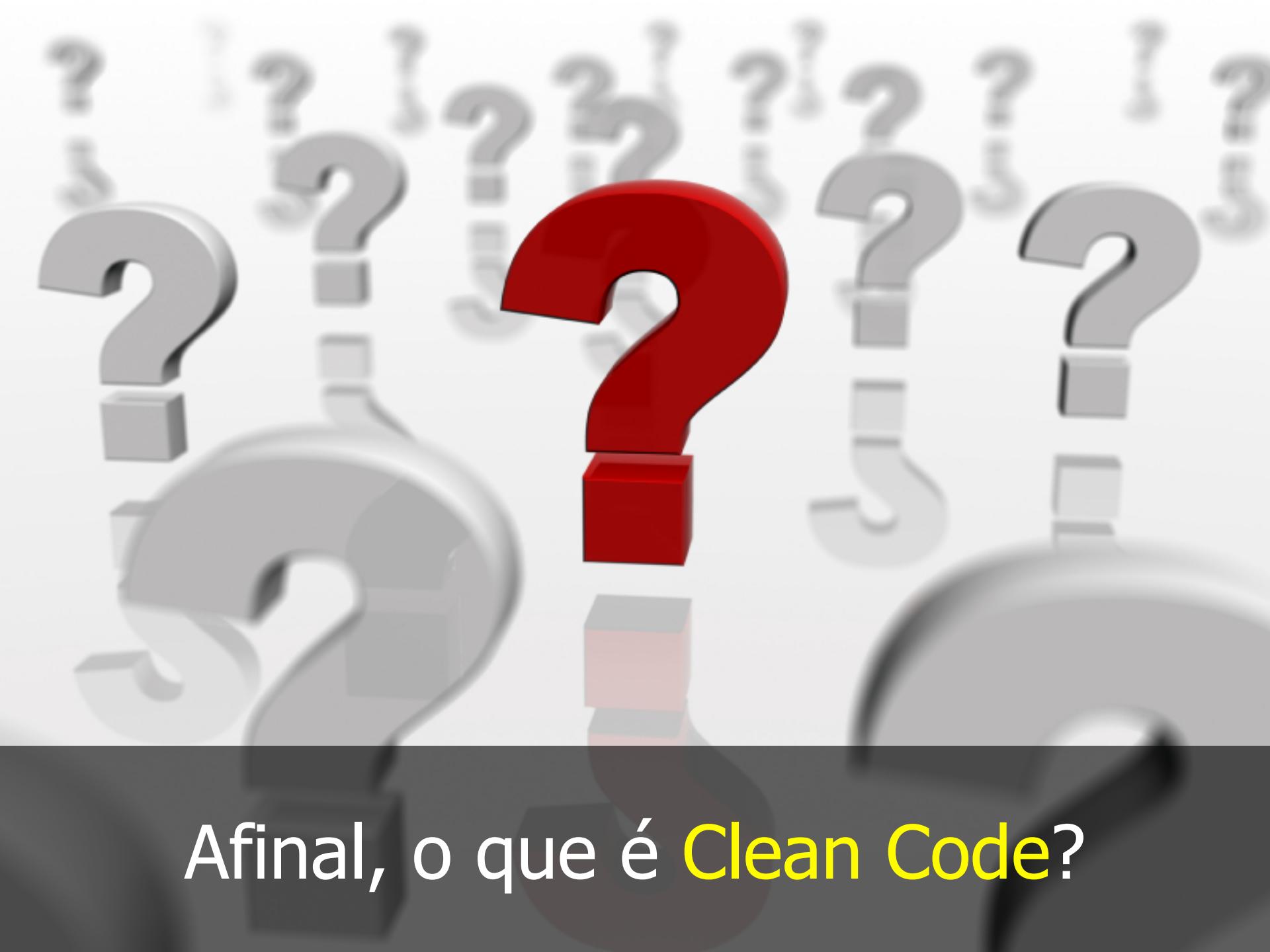
Fica cada vez mais difícil **evoluir** o produto



Com a concorrência da vez maior, o **comercial** começa a vender menos



O impacto financeiro aparece



Afinal, o que é Clean Code?

"Clean code is simple and direct"

Grady Booch (Criador da UML)

"Clean code always looks like it was written by someone who cares"

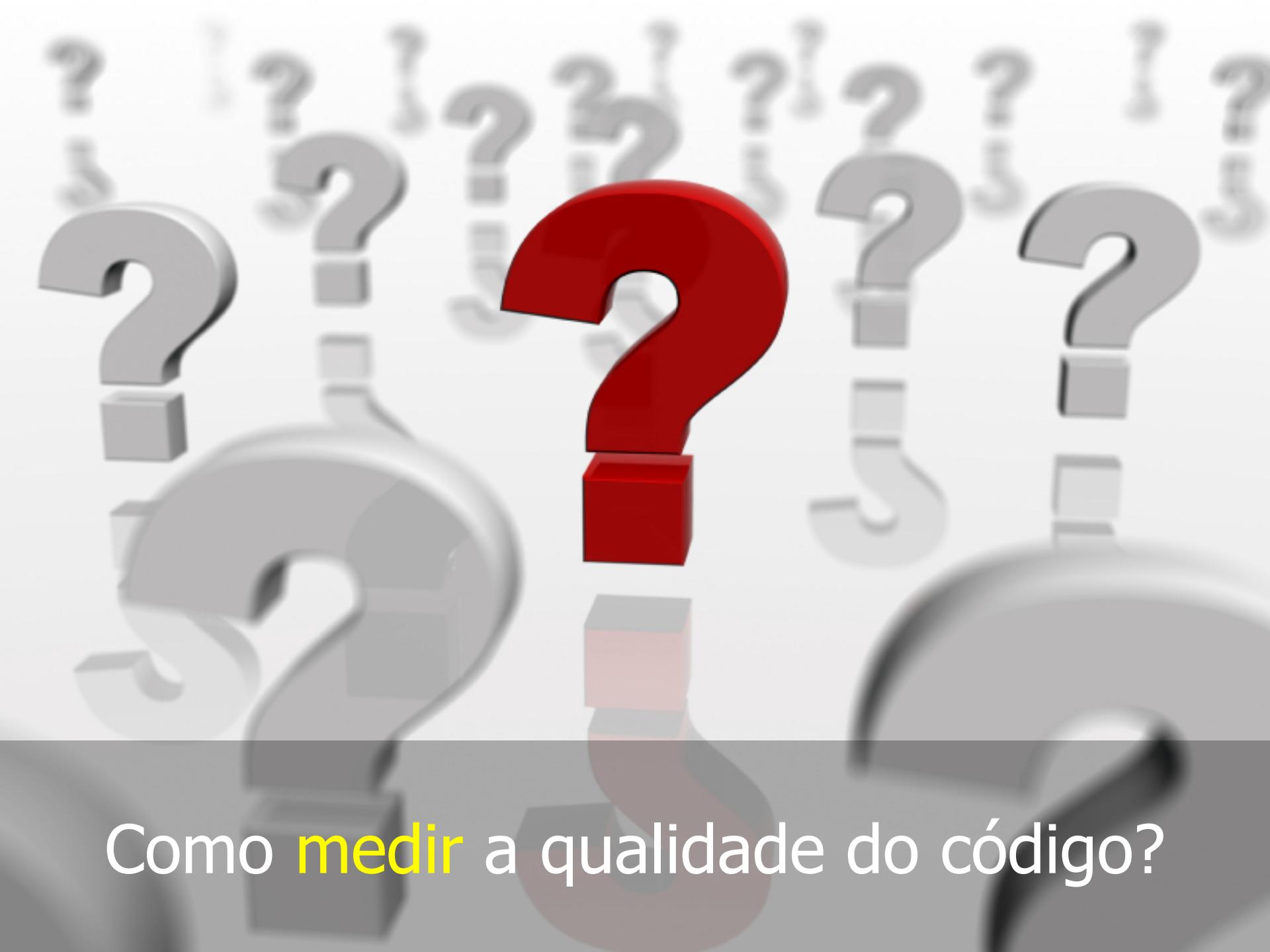
Michael Feathers (Autor do livro Working Effectively with Legacy Code)

"You know you are working on clean code when each routine you read turns out to be pretty much what you expected"

Ward Cunningham (Criador da Wiki)

"Any fool can write code that a computer can understand. Good programmers write code that humans can understand"

Martin Fowler (Um dos maiores autores sobre desenvolvimento de software)



Como **medir** a qualidade do código?

Linhas de código?

Número de métodos?

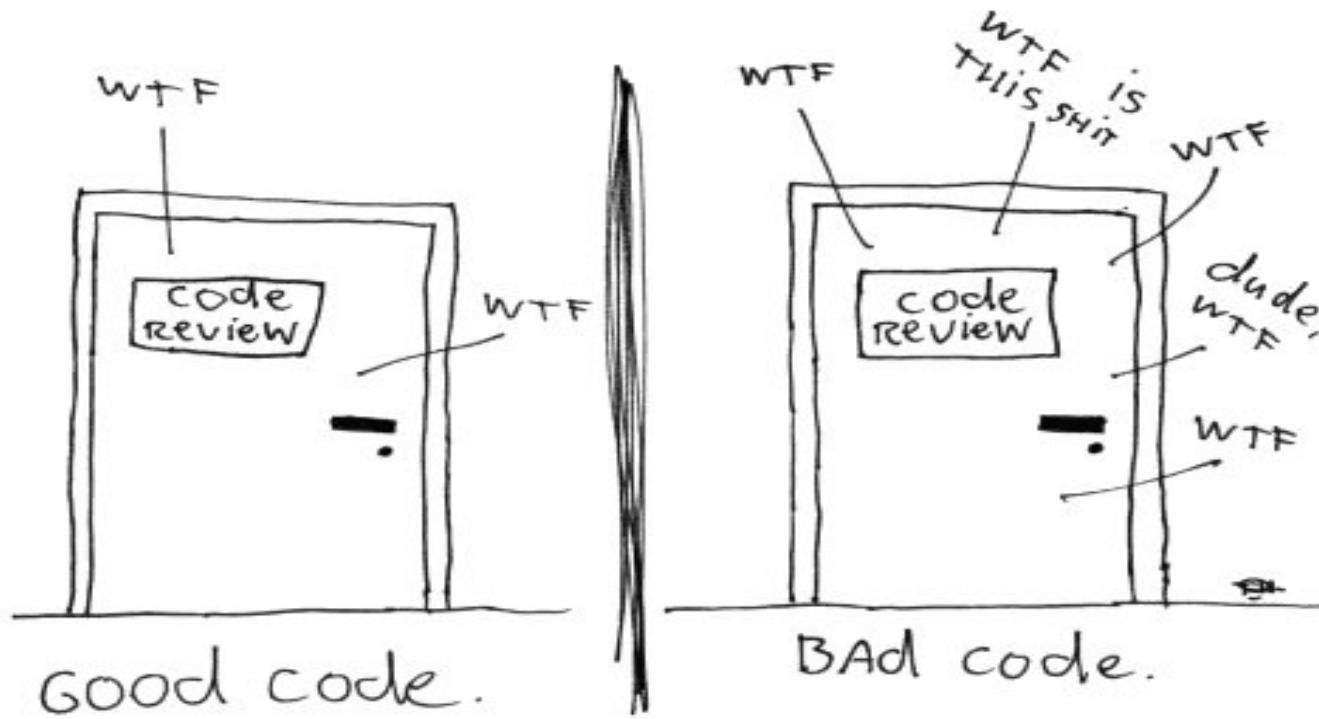
Número de classes?

Tamanho dos método?

Complexidade?

???
- - -

The ONLY VALID MEASUREMENT OF CODE QUALITY: WTFs/MINUTE



```
2445     if ((lowerChar == Character.ERROR) ||
2446         (lowerChar >= Character.MIN_SUPPLEMENTARY_CODE_POINT)) {
2447         if (lowerChar == Character.ERROR) {
2448             lowerCharArray =
2449                 ConditionalSpecialCasing.toLowerCaseCharArray(this, i, locale);
2450         } else if (srcCount == 2) {
2451             resultOffset += Character.toChars(lowerChar, result, i + resultOffset) - srcCount;
2452             continue;
2453         } else {
2454             lowerCharArray = Character.toChars(lowerChar);
2455         }
2456
2457         /* Grow result if needed */
2458         int mapLen = lowerCharArray.length;
2459         if (mapLen > srcCount) {
2460             char[] result2 = new char[result.length + mapLen - srcCount];
2461             System.arraycopy(result, 0, result2, 0,
2462                             i + resultOffset);
2463             result = result2;
2464         }
2465         for (int x=0; x<mapLen; ++x) {
2466             result[i+resultOffset+x] = lowerCharArray[x];
2467         }
2468         resultOffset += (mapLen - srcCount);
2469     } else {
2470         result[i+resultOffset] = (char)lowerChar;
2471     }
2472 }
2473 return new String(0, count+resultOffset, result);
2474 }
```

WTF?

```
2445 if ((lowerChar == Character.ERROR) ||
2446     (lowerChar >= Character.MIN_SUPPLEMENTARY_CODE_POINT)) {
2447     if (lowerChar == Character.ERROR) {
2448         lowerCharArray =
2449             ConditionalSpecialCasing.toLowerCaseCharArray(this, i, locale);
2450     } else if (srcCount == 2) {
2451         resultOffset += Character.toChars(lowerChar, result, i + resultOffset) - srcCount;
2452         continue;
2453     } else {
2454         lowerCharArray = Character.toChars(lowerChar);
2455     }
2456
2457     /* Grow result if needed */
2458     int mapLen = lowerCharArray.length;
2459     if (mapLen > srcCount) {
2460         char[] result2 = new char[result.length + mapLen - srcCount];
2461         System.arraycopy(result, 0, result2, 0,
2462                         i + resultOffset);
2463         result = result2;
2464     }
2465     for (int x=0; x<mapLen; ++x) {
2466         result[i+resultOffset+x] = lowerCharArray[x];
2467     }
2468     resultOffset += (mapLen - srcCount);
2469 } else {
2470     result[i+resultOffset] = (char)lowerChar;
2471 }
2472 }
2473 return new String(0, count+resultOffset, result);
2474 }
```

```
2445 if ((lowerChar == Character.ERROR) ||
2446     (lowerChar >= Character.MIN_SUPPLEMENTARY_CODE_POINT)) {
2447     if (lowerChar == Character.ERROR) {
2448         lowerCharArray =
2449             ConditionalSpecialCasing.toLowerCaseCharArray(this, i, locale);
2450     } else if (srcCount == 2) {
2451         resultOffset += Character.toChars(lowerChar, result, i + resultOffset) - srcCount;
2452         continue;
2453     } else {
2454         lowerCharArray = Character.toChars(lowerChar);
2455     }
2456
2457     /* Grow result if needed */
2458     int mapLen = lowerCharArray.length;
2459     if (mapLen > srcCount) {
2460         char[] result2 = new char[result.length + mapLen - srcCount];
2461         System.arraycopy(result, 0, result2, 0,
2462                         i + resultOffset);
2463         result = result2;
2464     }
2465     for (int x=0; x<mapLen; ++x) {
2466         result[i+resultOffset+x] = lowerCharArray[x];
2467     }
2468     resultOffset += (mapLen - srcCount);
2469 } else {
2470     result[i+resultOffset] = (char)lowerChar;
2471 }
2472 }
2473 return new String(0, count+resultOffset, result);
2474 }
2475 }
```

WTF?

WTF?

```
2445 if ((lowerChar == Character.ERROR) ||
2446     (lowerChar >= Character.MIN_SUPPLEMENTARY_CODE_POINT)) {
2447     if (lowerChar == Character.ERROR) {
2448         lowerCharArray =
2449             ConditionalSpecialCasing.toLowerCaseCharArray(this, i, locale);
2450     } else if (srcCount == 2) {
2451         resultOffset += Character.toChars(lowerChar, result, i + resultOffset) - srcCount;
2452         continue;
2453     } else {
2454         lowerCharArray = Character.toChars(lowerChar);
2455     }
2456
2457     /* Grow result if needed */
2458     int mapLen = lowerCharArray.length;
2459     if (mapLen > srcCount) {
2460         char[] result2 = new char[result.length + mapLen - srcCount];
2461         System.arraycopy(result, 0, result2, 0,
2462                         i + resultOffset);
2463         result = result2;
2464     }
2465     for (int x=0; x<mapLen; ++x) {
2466         result[i+resultOffset+x] = lowerCharArray[x];
2467     }
2468     resultOffset += (mapLen - srcCount);
2469 } else {
2470     result[i+resultOffset] = (char)lowerChar;
2471 }
2472 }
2473 return new String(0, count+resultOffset, result);
2474 }
2475
```

WTF?

WTF?

WTF?

WTF?







Muitas vezes, existe um ponto de não retorno,
evite chegar lá, poderá ser muito caro e muito
arriscado fazer qualquer mudança



Refatore antes que seja tarde demais