O que eu preciso para ser um excelente Engenheiro de Software?

Prof. Marco Tulio Valente

mtov@dcc.ufmg.br

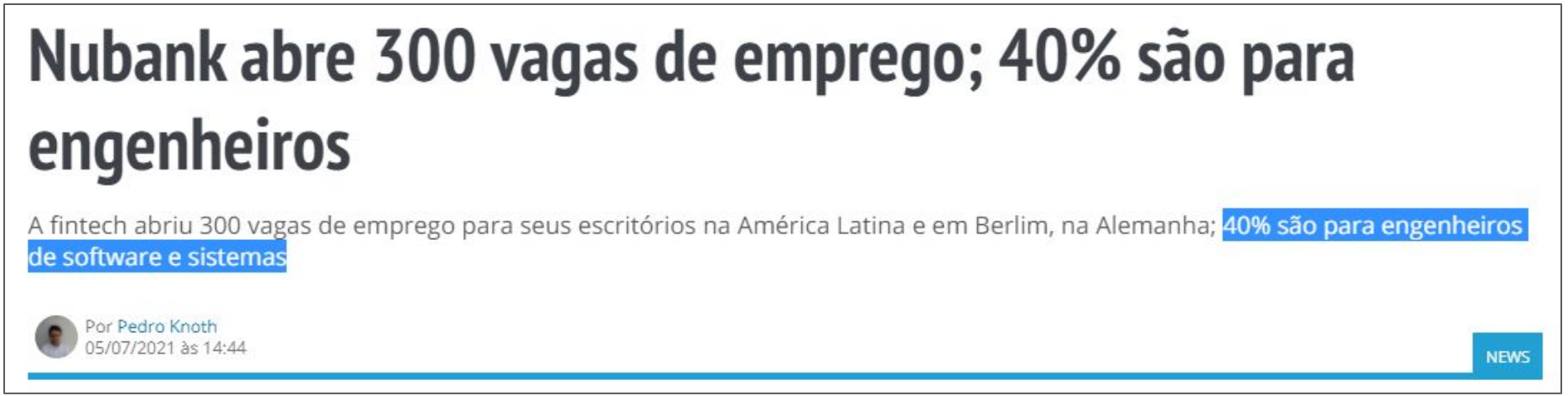
ERES 2024

Por que o Inter, Nubank, etc são **bancos digitais**?

Empresa digital ⇒ movida por software

Empresa de software ⇒ movida por **engenheiros de software**

GitLab team photo, New Orleans, 2019



Nubank abre 300 vagas de emprego; 40% são para engenheiros

A fintech abriu 300 vagas de emprego para seus escritórios na América Latina e em Berlim, na Alemanha; 40% são para engenheiros de software e sistemas

Por Pedro Knoth
05/07/2021 às 14:44

NEWS

https://tecnoblog.net/457977/nubank-abre-300-vagas-de-emprego-40-sao-para-engenheiros/

Toda empresa está se transformando em uma empresa de software

Inclusive indústrias mais tradicionais…

How Software Is Eating the Car

The trend toward self-driving and electric vehicles will add hundreds of millions of lines of code to cars. Can the auto industry cope?

By **Robert N. Charette**

100+ MLOC

IEEE Spectrum, Junho 2021

No futuro, existe uma boa chance de você ser um Engenheiro de Software!

Cargos de mercado para ES

Engenheiro de Software
Desenvolvedor
Desenvolvedor FullStack
Desenvolvedor Frontend
Desenvolvedor Backend
Desenvolvedor Mobile
Programador
Tech Lead
Arquiteto de Software

Scrum Master
Agile Master
Gerente de Projetos

Analista de Qualidade (QA)
Analista de Testes

Product Owner (PO)
Product Manager (PM)
Analista de Requisitos

Devs trabalham próximos dos seguintes cargos

UX/UI	Dados	Ops
UX Designer	Cientista de Dados	Administrador de Rede
UX Researcher	Engenheiro de Dados	Site Reliability Engineer
UX Writer		Engenheiro de Release
		Analista de Suporte
		Analista de Segurança

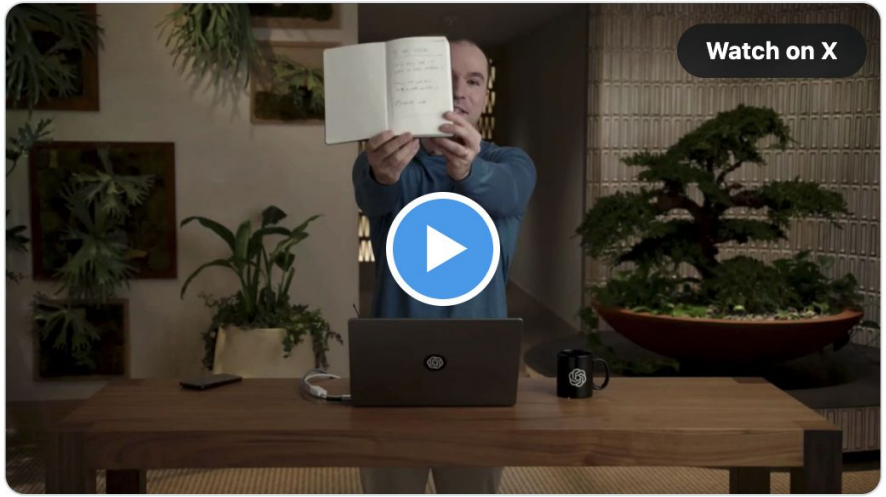
Mas a IA não vai acabar com a profissão de Engenheiro de Software?

Primeiro, existe muito hype…

Siqi Chen
@blader · **Follow**

gpt-4 can turn your napkin sketch into a web app, instantly.

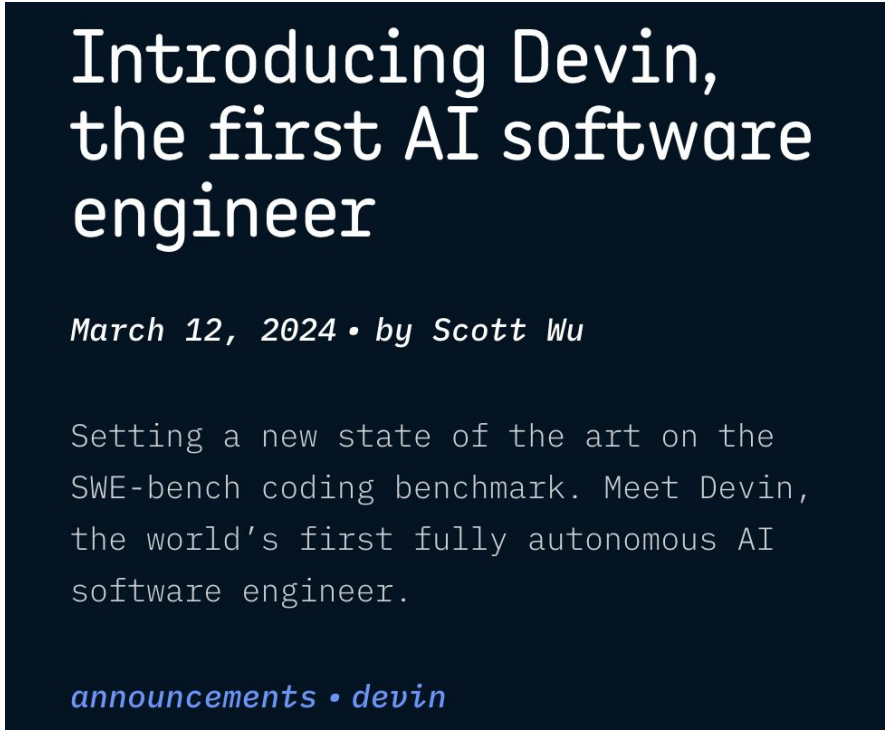
we are deep into uncharted territory here.



8:59 PM · Mar 14, 2023

4.5K Reply Copy link

Read 139 replies



https://www.cognition.ai/blog/introducing-devin

Google CEO says more than a quarter of the company's new code is created by AI

Hugh Langley Oct 29, 2024, 9:36 PM UTC

Share | Save

Google

https://archive.is/X43PU

I also work at google (until last Friday). Agree with what you said. My thoughts are

1. This quote is clearly meant to exaggerate reality, and they are likely including things like fully automated CL/PR's which have been around for a decade as "AI generated".

https://news.ycombinator.com/item?id=41991291

Claro, existem inúmeros exemplos de uso e de ganho de produtividade

Exemplo #1

Implemente um programa em Python que leia uma lista de CPFs de um arquivo texto, chamado cpfs.txt. Cada linha do arquivo contém um CPF.

O programa deve validar esses CPFs, de acordo com o dígito verificador de cada um deles.

No final, o programa deve imprimir na console os CPFs válidos e os inválidos.

```
def validar_cpf(cpf):
    cpf = ''.join(filter(str.isdigit, cpf))
    if len(cpf) != 11:
        return False
    if cpf == cpf[0] * 11:
        return False
    for i in range(9, 11):
        soma = sum((int(cpf[j]) * ((i + 1) - j) for j in range(i)))
        digito = (soma * 10) % 11
        if digito == 10:
            digito = 0
        if digito != int(cpf[i]):
            return False
    return True

def ler_cpfs_e_validar(nome_arquivo):
    with open(nome_arquivo, 'r') as arquivo:
        cpfs = arquivo.readlines()
    cpfs = [cpf.strip() for cpf in cpfs]
    cpfs_validos = [cpf for cpf in cpfs if validar_cpf(cpf)]
    cpfs_invalidos = [cpf for cpf in cpfs if not validar_cpf(cpf)]
    return cpfs_validos, cpfs_invalidos

def main():
    nome_arquivo = 'cpfs.txt'
    cpfs_validos, cpfs_invalidos = ler_cpfs_e_validar(nome_arquivo)
    print("CPFs Válidos:")
    for cpf in cpfs_validos:
        print(cpf)
    print("\nCPFs Inválidos:")
    for cpf in cpfs_invalidos:
        print(cpf)
```

Exemplo #2

Implemente um script em Shell que valide todos os links HTTP presentes em um arquivo de entrada no formato Markdown.

Basicamente, eu gostaria de saber se esses links estão funcionando ou se estão quebrados.

```
#!/bin/bash

# Check if filename is provided as a command-line argument
if [ $# -eq 0 ]; then
    echo "Usage: $0 <markdown_file>"
    exit 1
fi

# Extract filename from command-line argument
markdown_file=$1

# Step 1: Extract all links from the markdown document
links=$(grep -oE '\[.*\]\(http[s]?://[^)]+\)' "$markdown_file" | sed -E
's/.*\[(.*)\]\((http[s]?:\/\/[^)]+)\).*/\2/')

# Step 2: Loop through each link and check its validity using curl
for link in $links; do
    status=$(curl -s -o /dev/null -w "%{http_code}" -L $link)
    if [ $status -eq 200 ] || [ $status -eq 418 ]; then        echo "Link
$link is valid."
    else
        echo "Link $link is broken (HTTP status code: $status)."
    fi
    sleep 70  # seconds
done
```

Mas existem também dificuldades!

"O Copilot foi usado para o back-end e o ChatGPT para o front-end, aumentando a produtividade, contribuindo em códigos simples e para debug. Todavia, como utilizamos uma versão nova do Next, muitas coisas essas LLMs não sabiam, sendo necessário consultar a documentação."

Comentário de aluno da disciplina Engenharia de Software, DCC/UFMG

"O grupo fez uso das duas ferramentas, que na grande maioria das vezes ajudaram muito a explicar erros de código e como fazer alguns métodos. Notou-se, porém, que a ferramenta por muitas vezes atrapalhava o andamento de alguma tarefa por não entender o contexto geral do projeto e dava como resposta algum trecho de código que mais trazia erros do que respostas."

Comentário de aluno da disciplina Engenharia de Software, DCC/UFMG

"Diversas vezes, precisamos corrigir bugs que a IA gerou. Quando pedimos features mais complicadas, como a parte de estatísticas, a IA gerou um código bem confuso, usando variáveis que nem foram definidas ainda. Isso acabou nos exigindo mais tempo para poder analisar e corrigir."

Comentário de aluno da disciplina Engenharia de Software, DCC/UFMG

Mas qual a sua opinião?

Programação vs Engenharia de Software

Programação

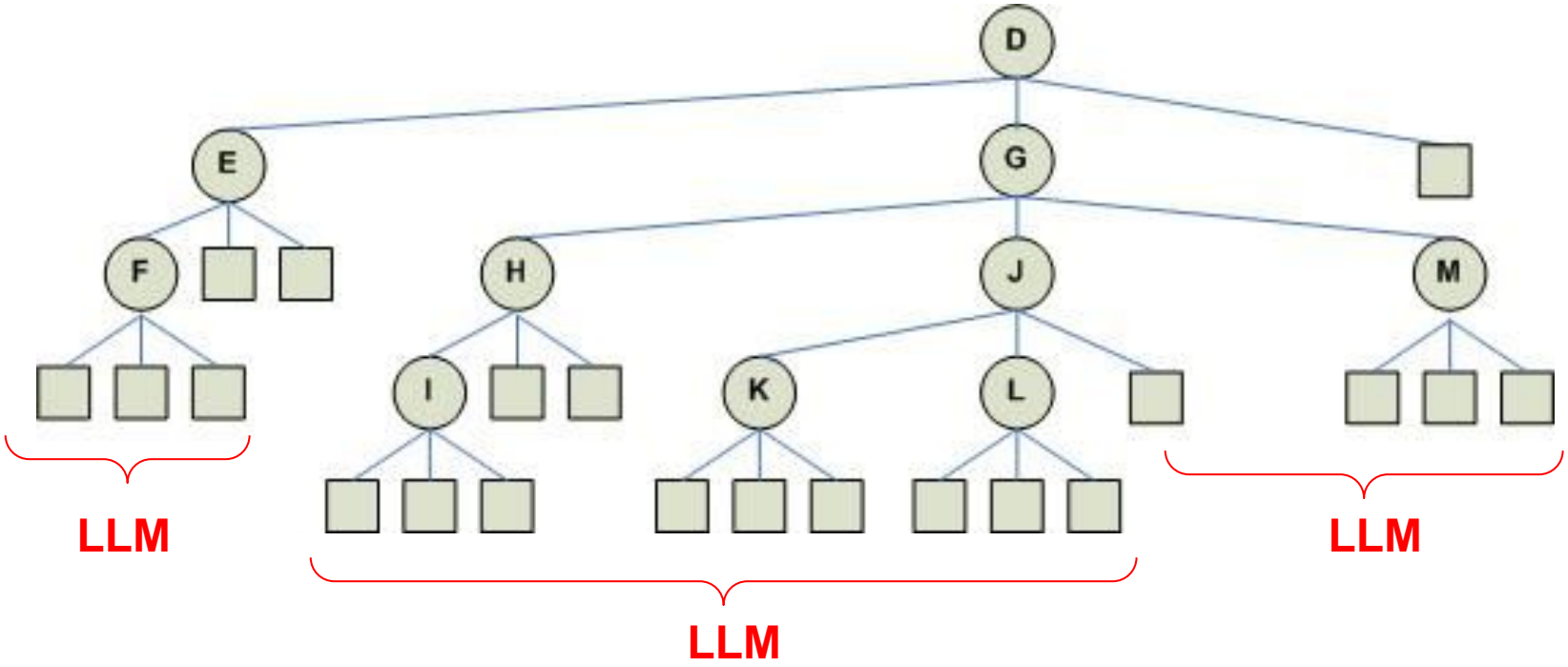
- Um problema, algoritmo ou funcionalidade
- Um desenvolvedor
- Poucas linhas de código (~KLOC)
- Pouca ou nenhuma manutenção futura

Engenharia de Software

- Dezenas de funcionalidades
- Milhões de linhas de código
- Dezenas de desenvolvedores
- Diversos clientes, versões e tecnologias
- Interfaces e integrações externas
- Manutenção e operação por anos
- Requisitos não-funcionais (segurança, escalabilidade, etc)
- etc

Quando LLMs se destacam?

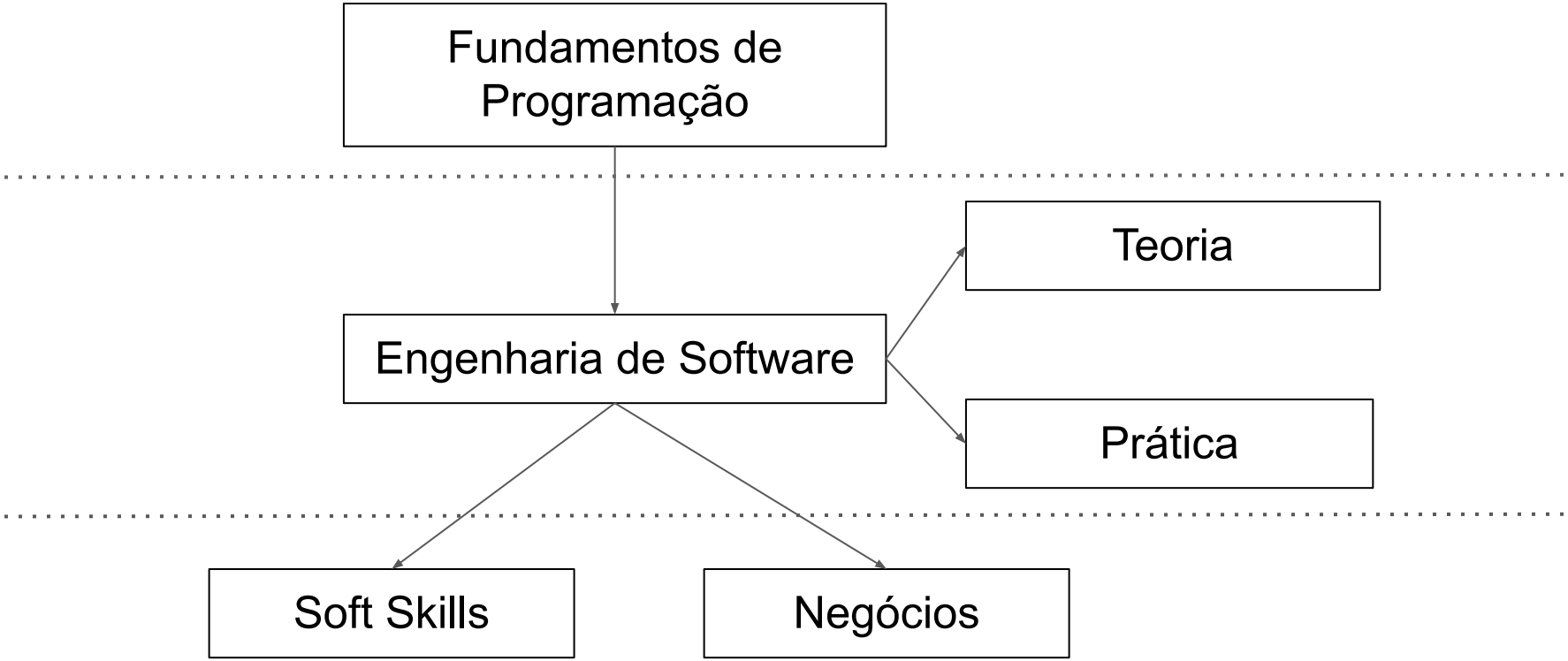
- Programação (programming in the small)
- Principalmente, geração de código mais comum e simples


D
E
G
F
H
J
M
I
K
L
LLM
LLM
LLM

Qual o problema LLMs não resolvem?

- Programming in the large (4Ts)
  - **T**amanho, ou seja, complexidade (menos "repetição)
  - **T**imes e pessoas
  - **T**empo (manutenção por anos)
  - **T**ecnologias variadas

Explicando melhor o trabalho de um Engenheiro de Software


Fundamentos de Programação
Engenharia de Software
Teoria
Prática
Soft Skills
Negócios

Camada 1: Fundamentos de Programação

- Todo engenheiro de software deve saber programar e gostar de programar




A Base

Camada 2.1: Teoria de Engenharia de Software

- Processos: Scrum, Kanban, XP, Shape Up, etc
- Requisitos: Histórias de Usuários, MVPs, Testes A/B, JTBD, DT
- Modelos: UML
- Projeto: coesão, acoplamento, SOLID, padrões de projeto
- Arquitetura: MVC, Limpa, Hexagonal, Microsserviços, Pub/Sub
- Testes: unidade, integração, end-to-end, snapshot, etc
- Refactoring
- DevOps: controle de versões, CI/CD, etc

Físico

- Complexidade visível
- Mudanças menos frequentes

- Projeto up-front
- Entrega única no final

Engenharias Tradicionais

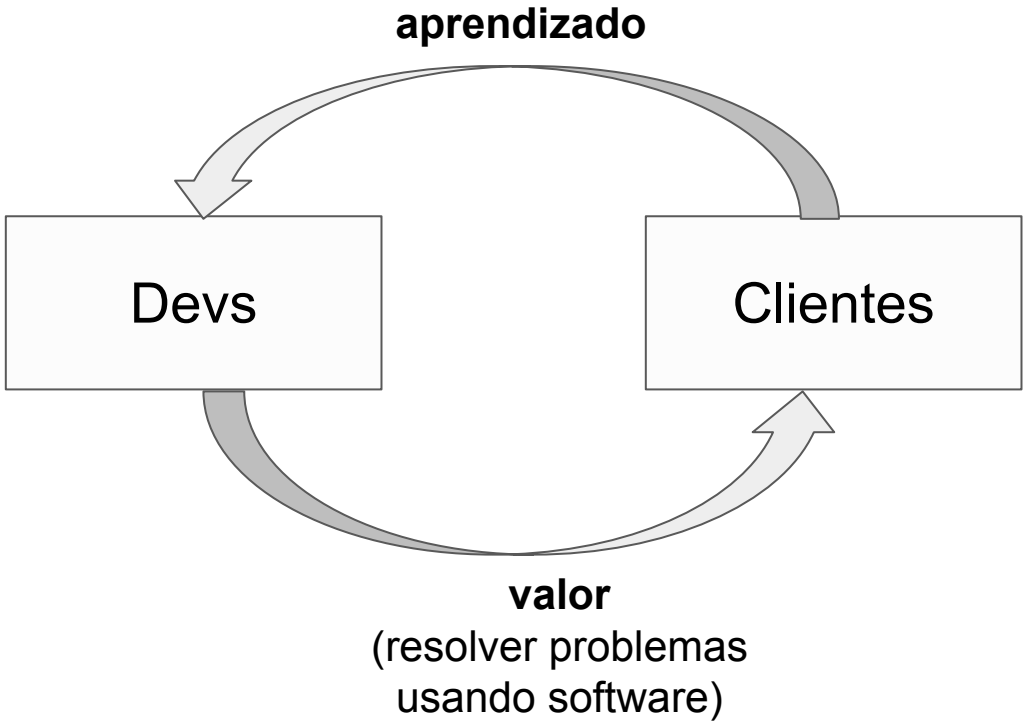
Digital

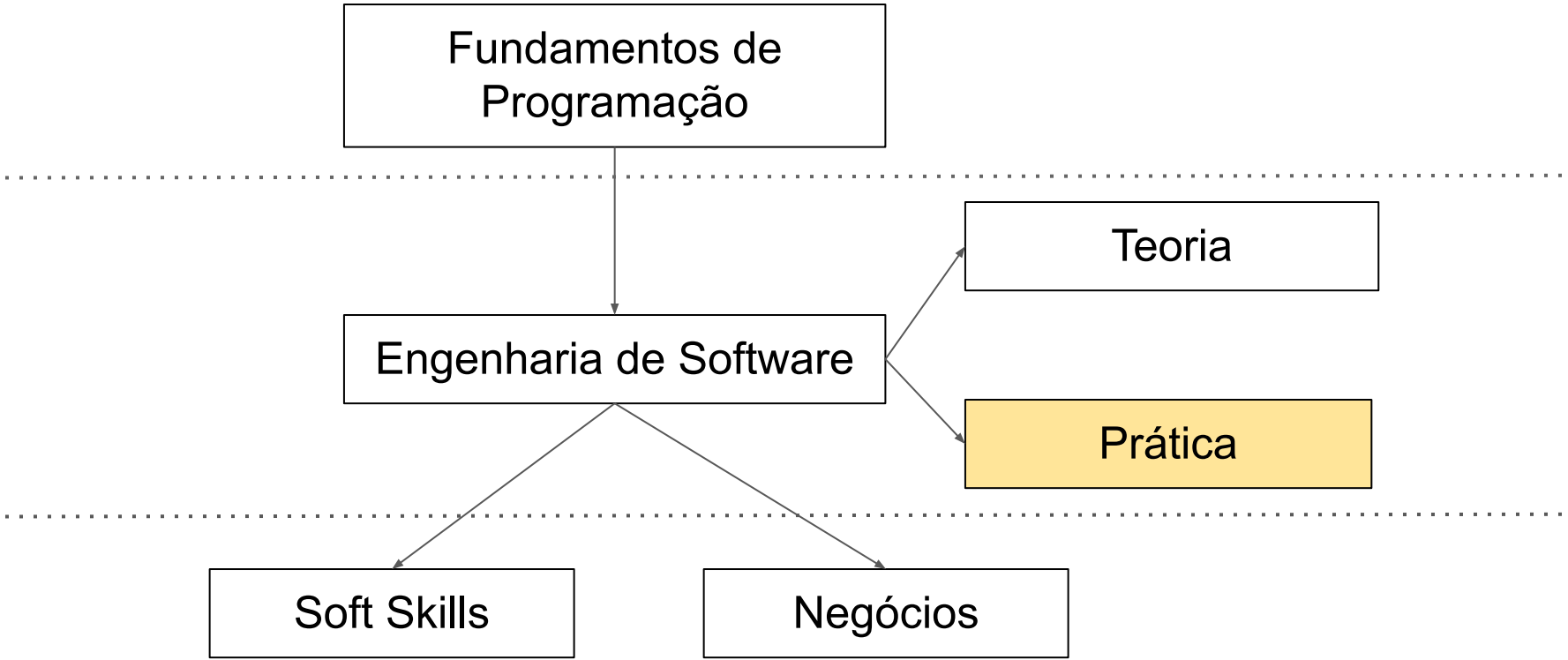
- Complexidade invísivel, abstrata
- Mudanças muito mais frequentes

- Projeto evolucionário
- Entrega contínua

Engenharia de Software (Agilidade)

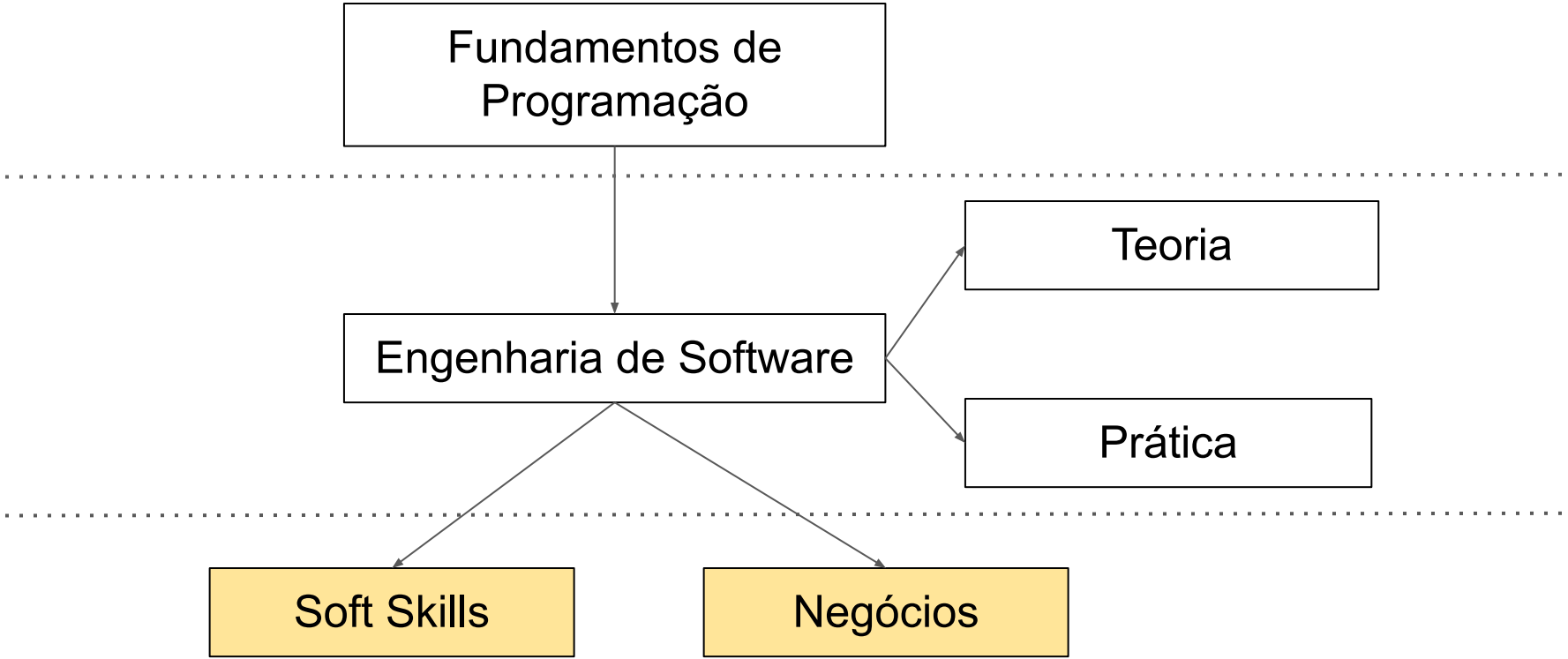
Modelo Mental: agilidade




Fundamentos de Programação
Engenharia de Software
Teoria
Prática
Soft Skills
Negócios

Camada 2.2: Prática em Engenharia de Software

- Não adianta a “teoria”, se você não entregar resultado
- Resultado = “software em funcionamento”
- Tecnologia: linguagens, stacks, frameworks, ferramentas, etc

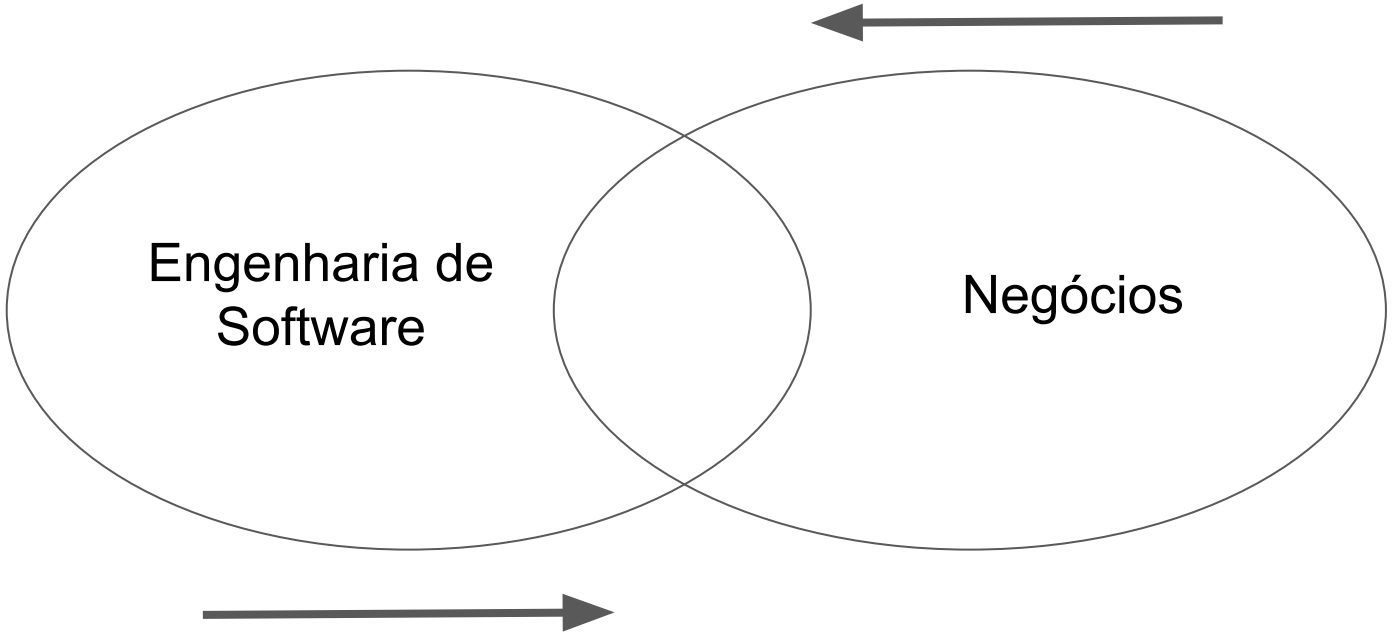

Fundamentos de Programação
Teoria
Engenharia de Software
Prática
Soft Skills
Negócios

Camada 3.1: Soft/People Skills

- Empresa de Software = pessoas
- Soft ou people skills são muito importantes!
- Intrapessoais: autoconfiança, resiliência, flexibilidade, adaptabilidade, proatividade, compromisso, autoconfiança
- Interpessoais: trabalho em equipe, liderança, capacidade de negociação, comunicação, oratória

“Empresas contratam por hard skills, mas demitem por soft skills”

Camada 3.2: Negócios



X = empresa & app

X = Uber, iFood, Zoom, YouTube, Inter, Nubank, etc.

“Eu não quero desenvolvedor só tarefeiro…”

Obrigado!

mtov@dcc.ufmg.br