Inglês Técnico

Engenharia de Software

Sumário

Introdução

Neste material, exploraremos os termos e expressões essenciais necessários para uma comunicação eficaz no campo do desenvolvimento de software.

Você desenvolverá habilidades para articular conceitos e técnicas de forma clara e precisa, enquanto adquire a capacidade de compreender e responder de maneira eficaz aos desafios e demandas da engenharia de software.

Estamos entusiasmados para iniciar esta jornada de aprendizado com você, contribuindo para o seu sucesso profissional como Engenheiro de Software!

 Vamos começar!

Orientação

Siga estas orientações para otimizar o uso deste material e potencializar o seu aprendizado

1 Imprima este PDF;

2 Destaque com caneta marca-texto as palavras desconhecidas;

3 Leia a coluna *meaning* para descobrir o significado, sem usar tradutores;

4 Construa frases com aplicação das novas palavras que você está aprendendo.

Se precisar de inspiração, use o **dictionary.cambridge.org**

Faça isso por meio da escrita e não da digitação, pois isso potencializa o armazenamento do novo conhecimento na memória de longo prazo.

Expressões

Exemplos
Computer-aided design Desenho auxiliado por computador	**The latest computer-aided design systems are used to ensure speed and accuracy.** Os mais recentes sistemas de desenho auxiliado por computador são usados para garantir velocidade e precisão.
Continuous Integration Integração continua	**Continuous Integration is one of the first things you should do in a software development cycle.** A integração contínua é uma das primeiras coisas que você deve fazer em um ciclo de desenvolvimento de software.
Cross-site scripting attacks Ataques de script entre sites	**Such files are accessible via HTTP and can be used for cross-site scripting attacks or disclose information unintendedly.** Esses arquivos podem ser acessados via HTTP e podem ser usados para ataques de script entre sites ou divulgar informações não-intencionalmente.
Data encapsulation Encapsulamento de dados	**Data encapsulation and multi-tiered design go hand in hand.** O encapsulamento de dados e o design em várias camadas andam de mãos dadas.
Data structure Estrutura de dados	**You need to add one component to the application, but the rest of your data structure doesn't need to be touched.** Você precisa adicionar um componente ao aplicativo, mas o resto da estrutura de dados não precisa ser alterado.
Direct memory access Acesso direto à memória	**Last but not least, define guidelines for direct memory access channel and possible threats and solutions.** Por último, mas não menos importante, defina diretrizes para o canal de acesso direto à memória e possíveis ameaças e soluções.
Dynamic programming Programação dinâmica	**I think there are some issues with reflection and dynamic programming that need to be cleaned up.** Acho que há alguns problemas com reflexão e programação dinâmica que precisam ser resolvidos.
Encapsulation Encapsulamento	**Encapsulation makes virtual machines incredibly portable and easy to manage.** O encapsulamento torna as máquinas virtuais incrivelmente portáteis e fáceis de gerenciar.
End-user Usuário final	**The end-user only needs to deal with information retrieval within a limited area.** O usuário final precisa apenas lidar com a recuperação de informações dentro de uma área limitada.
Fail-safe À prova de falhas	**It is important to have a fail-safe system, and that's why I've decided to build mine with a RAID.** É importante ter um sistema à prova de falhas e é por isso que decidi construir o meu com RAID.

Expressões

Exemplos	
Functional design Design funcional	**There are quite a few mistakes in the functional design of the entire data exchange system.** Existem alguns erros no design funcional de todo o sistema de troca de dados.
Functional programming Programação funcional	**It also offers good support for object-oriented programming, functional programming, and data-driven programming.** Ele também oferece um bom suporte para programação orientada a objetos, programação funcional e programação orientada a dados.
Greedy algorithm Algoritmo guloso	**It is, of course, very easy to improve the efficiency of the greedy algorithm.** Obviamente, é muito fácil melhorar a eficiência do algoritmo guloso.
Hash table Tabela Hash ou de Dispersão	**This method uses a pre-computed hash table to serve as an index for short sequences.** Este método usa uma tabela de hash pré-calculada para servir como um índice para sequências curtas.
Interoperability Interoperabilidade	**You are using a beta version of the interoperability feature.** Você está usando uma versão beta do recurso de interoperabilidade.
Iterative model Modelo iterativo	**The iterative model is an important method to develop reusable software.** O modelo iterativo é um método importante para desenvolver software reutilizável.
Legacy software Software legado	**Legacy software can have embedded text and graphics which you may have always wanted to change.** O software legado pode ter texto e gráficos incorporados que você sempre quis alterar.
Non-functional requirements Requisitos não funcionais	**Tests should also assess non-functional requirements such as robustness, availability and performance.** Os testes também devem avaliar os requisitos não funcionais, como robustez, disponibilidade e desempenho.
Off-the-shelf software Software de prateleira	**You may decide to evaluate off-the-shelf software for particularly risky elements.** Você pode decidir avaliar o software de prateleira para elementos particularmente arriscados.
Plugability Conectividade	**It enables quick and easy coding thanks to its user-friendly pluggability.** Ele permite uma codificação rápida e fácil graças à sua conectividade amigável.

Expressões

Exemplos	
Read-only memory Memória somente leitura	The most important programs are in the read-only memory. Os programas mais importantes estão na memória somente leitura.
Static variables Variáveis estáticas	This process can be a lifesaver if your application relies heavily on static variables. Este processo pode ser um salva-vidas se o seu aplicativo depender muito de variáveis estáticas.
Stepwise refinement Refinamento gradual	Through stepwise refinement, I can set forth the broad outlines of a procedure first and only later fill in the details. Por meio de um refinamento gradual, posso definir os contornos gerais de um procedimento primeiro e só depois preencher os detalhes.
Traceability Rastreabilidade	Our traceability systems are probably some of the best in the world. Nossos sistemas de rastreabilidade são provavelmente alguns dos melhores do mundo.
Waterfall model Modelo em cascata	The standard tool often forces a waterfall model, and frankly, waterfall is hideous. A ferramenta padrão geralmente força um modelo em cascata e, francamente, a cascata é horrível.

Glossário

A

Agile software development	Calls for keeping code simple, testing often, and delivering small, functional bits of the application as soon as they're ready. The focus is to build a succession of parts, rather than delivering one large application at the end of the project.
Amdahl's law	Stipulates that, in a program with parallel processing, a relatively few instructions that have to be performed in sequence will have a limiting factor on program speedup such that adding more processors may not make the program run faster.
Amelioration pattern	A design pattern that describes how to go from a bad solution to a better one.
Antipattern	A frequently used, but largely ineffective solution to a problem. The term was originally used to refer to a design pattern gone wrong.
Application programming interface (API)	A specific method prescribed by a computer operating system or by an application program by which a programmer writing an application program can make requests of the operating system or another application.
Application integration	The process of bringing data or a function from one application program together with that of another application program. Where these programs already exist, the process is sometimes realized by using middleware.
Application program	A program designed to perform a specific function directly for the user or, in some cases, for another application program.
Aspect-oriented programming (AOP)	An approach to programming that allows global properties of a program to determine how it is compiled into an executable program.

Glossário

B

Best practice	A technique or methodology that, through experience and research, has proven to reliably lead to a desired result.
Bug	A coding error in a computer program.
Build	A version of a program, usually pre-release, and identified by a build number, rather than by a release number. As a verb, to build can mean either to write code or to put individual coded components of a program together.
Build tool	A programming utility that is used when building a new version of a program.

C

Capability Maturity Model	A methodology used to develop and refine an organization's software development process. The model describes a five-level evolutionary path of increasingly organized and systematically more mature processes.

D

Data modeling	The analysis of data objects that are used in a business or other context and the identification of the relationships among these data objects.
Debugging	The process of locating and fixing or bypassing bugs (errors) in computer program code or the engineering of a hardware device.
Design pattern	A written document that describes a general solution to a design problem that recurs repeatedly in many projects.
Development environment	The set of processes and programming tools used to create the program or software product.
Development process	A set of tasks performed for a given purpose in a software development project.

Glossário

Driver	A program that interacts with a particular device or special kind of software. The driver contains special knowledge of the device or special software interface that programs using the driver do not.
Driver development kit (DDK)	A set of programs and related files that are used to develop a new software or hardware driver or to update an existing legacy application driver for an operating system.

E

Elegant solution	A solution in which the maximum desired effect is achieved with the smallest, or simplest effort.
Embedded systems programming	The programming of an embedded system in some device using the permitted programming interfaces provided by that system.
Enterprise application integration	The plans, methods, and tools aimed at modernizing, consolidating, and coordinating the computer applications in an enterprise.
Entity-relationship diagram	A data modeling technique that creates a graphical representation of the entities, and the relationships between entities, within an information system.
Ergonomics	The science of refining the design of products to optimize them for human use. Human characteristics, such as height, weight, and proportions are considered, as well as information about human hearing, sight, temperature preferences, and so on.
Exploratory model	A systems development method that consists of planning and trying different designs until one of them seems to be the right one to develop.
Extreme Programming	A pragmatic approach to program development that emphasizes business results first, and takes an incremental, get-something-started approach to building the product, using continual testing and revision.

Glossário

F

Feature creep	A tendency for product or project requirements to increase during development beyond those originally foreseen, leading to features that weren't originally planned and resulting risk to product quality or schedule.
Functional programming	A style of programming that emphasizes the evaluation of expressions rather than the execution of commands.
Functional specification	A formal document used to describe in detail for software developers a product's intended capabilities, appearance, and interactions with users.

G

Gantt chart	A horizontal bar chart frequently used in project management that provides a graphical illustration of a schedule that helps to plan, coordinate, and track specific tasks in a project.
Gap analysis	The study of the differences between two different information systems or applications, often for the purpose of determining how to get from one state to a new state. Sometimes spoken of as "the space between where we are and where we want to be."
Genetic programming	A model of programming which uses the ideas of biological evolution to handle a complex problem, most appropriate with problems in which there are a large number of fluctuating variables, such as those related to artificial intelligence.
Gold code	The final, ready-to-manufacture (that is, replicate onto media) version of the software.

H

Help system	A documentation component of a software program that explains the features of the program and helps the user understand its capabilities.
Hotfix	Code (sometimes called a patch) that fixes a bug in a product.
Human factors	The study of how humans behave physically and psychologically in relation to particular environments, products, or services.

Glossário

I

Information architecture	The set of ideas about how all information in a given context should be treated philosophically and, in a general way, how it should be organized; this is expressed in an information architecture document.
Information design	The detailed planning of specific information that is to be provided to a particular audience to meet specific objectives. In one hierarchical model, the information design follows the information architecture and information planning stages.
Integrated development environment	A programming environment that has been packaged as an application program, typically consisting of a code editor, a compiler, a debugger, and a GUI builder.
Independent software vendor (ISV)	A company that makes and sells software products that run on one or more computer hardware or operating system platforms.
Iterative	Describes a heuristic planning and development process where an application is developed in small sections called iterations.
ITIL	A set of best practices standards for information technology (IT) service management developed by the United Kingdom's Central Computer and Telecommunications Agency (CCTA).

J

Joint application development	A methodology that involves the client or end user in the design and development of an application, through a succession of collaborative workshops called JAD sessions.

K

KISS Principle (Keep it simple, stupid)	The principle that people want products that are easy to learn and use, and that companies realize time and cost benefits by producing such products.

Glossário

L

Lean programming	A concept that emphasizes optimizing efficiency and minimizing waste in the development of a computer program; the concept is also applicable to all enterprise practices.
Legacy application	An enterprise application that is based on languages, platforms, and/or techniques that predate current technology.

M

Metric	The measurement of a particular characteristic of a program's performance or efficiency.

O

Object-oriented programming	A programming model organized around objects rather than actions and data rather than logic, based on the idea that what we really care about are the objects we want to manipulate, rather than the logic required to manipulate them.
Open source	Describes a program whose source code is made available for use or modification as users or other developers see fit.
Outsourcing	An arrangement in which one company provides services for another company that could also be or usually have been provided in-house.

P

Pasta Theory of Programming	The idea that various programming structures can be likened to the structures of well-known pasta dishes: unstructured procedural programming is called spaghetti code , structured programming is called lasagna code , and object-oriented programming is called ravioli code.
Patch	A quick-repair job for the problems in a piece of programming, often available for download through the software maker's Web site.
Peer review	A process used for checking the work performed by one's equals (peers) to ensure it meets specific criteria.

Glossário

Program Evaluation Review Technique (PERT chart)	A project management tool used to schedule, organize, and coordinate tasks within a project developed by the U.S. Navy in the 1950s.
Polymorphism	From the Greek meaning "having multiple forms," the characteristic of being able to assign a different meaning or usage to something in different contexts - specifically, to allow an entity such as a variable, a function, or an object to have more than one form.
Portability	A characteristic attributed to a computer program if it can be used in an operating systems other than the one in which it was created without requiring major rework.
PRINCE2	A project management methodology developed by the government of the United Kingdom that makes use of the best proven practices from a variety of industries and backgrounds.
Program layer	A separate functional component that interacts with others in some sequential and hierarchical way, with each layer usually having an interface only to the layer above it and the layer below it.
Project planning	A discipline for stating how to complete a project within a certain timeframe, usually with defined stages, and with designated resources.
Prototyping	A systems development method (SDM) in which a prototype (an early approximation of a final system or product) is built, tested, and then reworked as necessary until an acceptable prototype is finally achieved from which the complete system or product can now be developed.
Pseudocode	A detailed yet readable description of what a computer program or algorithm must do, expressed in a formally-styled natural language rather than in a programming language. (Pronounced SOO-doh-kohd)

Glossário

R

Rapid application development (RAD)	Believe it or not, this means "a quick overview". Why anybody felt that describing it in this way was necessary is anybody's guess; but the same could be said of any of the examples on this list of office jargon, so you're best off avoiding these terms if you want to be taken seriously in an office environment.
Rational Unified Process (RUP)	An object-oriented and Web-enabled program development methodology that is said to be like an online mentor that provides guidelines, templates, and examples for all aspects and stages of program development.
Refactoring	A process that improves the internal structure of a software system without changing its external behavior.
Regression testing	The process of testing changes to computer programs to make sure that the older programming still works with the new changes.
Risk management	The process of planning, organizing, leading, and controlling the activities of an organization in order to minimize the effects of risk on an organization's capital and earnings.
Runtime	When a program is running.

S

Software development kit (SDK)	A set of programs used by a computer programmer to write application programs.
Service pack	An orderable or downloadable update to a customer's software that fixes existing problems and, in some cases, delivers product enhancements.
Shotgun debugging	The debugging of a program, hardware, or system problem using the approach of trying several possible solutions at the same time in the hope that one of them will work.

Glossário

Smoke testing	Non-exhaustive software testing, ascertaining that the most crucial functions of a program work, but not bothering with finer details.
Spaghetti code	Computer programming that is unnecessarily convoluted, and particularly programming code that uses frequent branching from one section of code to another.
Spiral model	A systems development method (SDM) that combines the features of the prototyping model and the waterfall model.
Structured Systems Analysis & Design Method (SSADM)	A widely-used computer application development method in the UK that divides an application development project into modules, stages, steps, and tasks, and provides a framework for describing projects in a fashion suited to managing the project.
Structured programming	A subset of procedural programming that enforces a logical structure on the program being written to make it more efficient and easier to understand and modify.
Synchronize-and-stabilize	A systems development life cycle model in which teams work in parallel on individual application modules, frequently synchronizing their code with that of other teams, and debugging (stabilizing) code regularly throughout the development process.
Systems development method - (SDM)	A work discipline that is chosen by the developers of a computer system or product as a way to ensure successful results.
Systems development life cycle model (SDLC)	One of a number of structured approaches to information system development, created to guide all the processes involved, from an initial feasibility study through maintenance of the completed application. Models include the waterfall model; rapid application development (RAD); joint application development (JAD); the fountain model; the spiral model; build and fix; and synchronize-and-stabilize.
Systems thinking	A holistic approach to analysis that focuses on the way that a system's constituent parts interrelate and how systems work over time and within the context of larger systems.

Glossário

T

Total cost of ownership (TCO)	A type of calculation designed to help consumers and enterprise managers assess both direct and indirect costs and benefits related to the purchase of any IT component.
Tool Kit (Tk)	A companion program to Tool Command Language (Tcl) for creating graphical user interfaces. Together with Tcl, Tk is a rapid program development tool.

U

User acceptance testing	A phase of software development in which the software is tested in the "real world" by the intended audience.
User interface	Everything designed into an information device with which a human being may interact -- including display screen, keyboard, mouse, light pen, the appearance of a desktop, illuminated characters, help messages, and how an application program or a Web site invites interaction and responds to it.
Utility	A small program that provides an addition to the capabilities provided by the operating system.

W

Waterfall model	Popular version of the systems development life cycle model that describes a linear and sequential development method.
Web services	Services made available from a business's Web server for Web users or other Web-connected programs.
Write-only code	Programming code that is hard to read.

Inglês Corporativo

Explore estes recursos para expandir seu vocabulário corporativo:


Rotina de trabalho em inglês
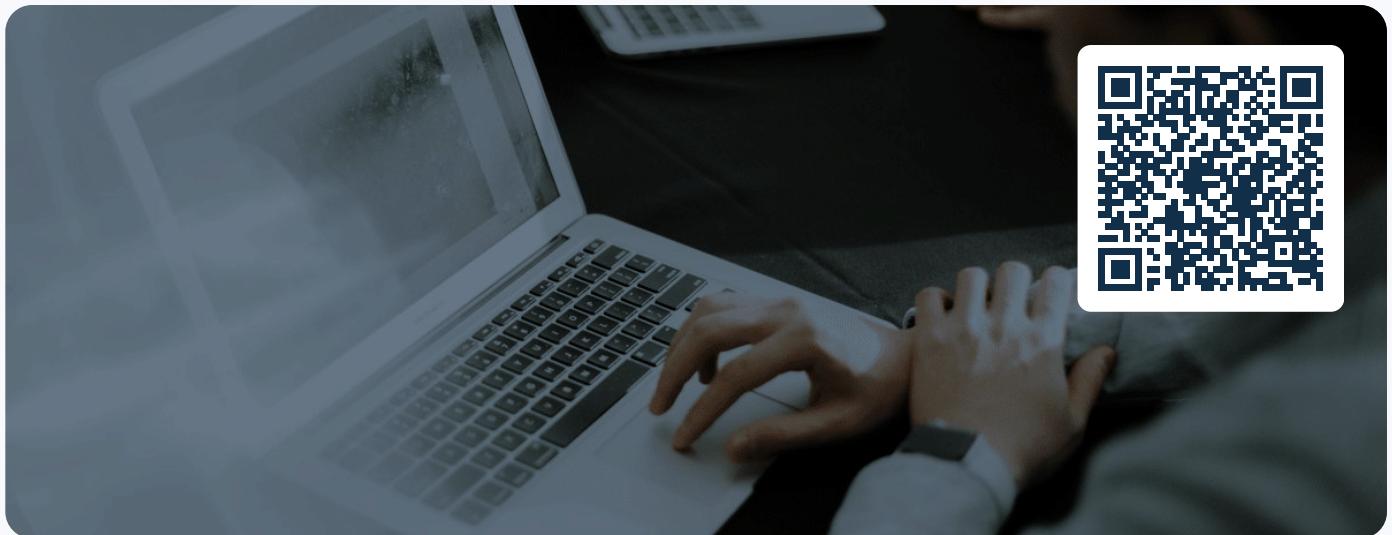
**Inglês corporativo técnico:
vocabulário útil para sua profissão**

Curso Fluência Descomplicada

Se você busca confiança e fluência no inglês, o Curso Fluência Descomplicada é o caminho ideal!

Com um método focado na prática, nele você participa de encontros online para treinar conversação, tira dúvidas ao vivo com professores e aprende estratégias eficazes para acelerar seu aprendizado
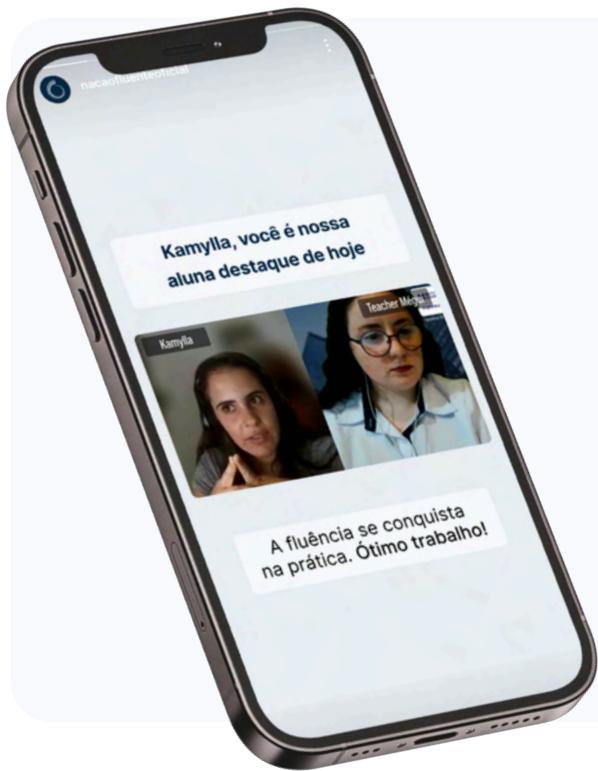
O que você encontra no curso?

- **Encontros de conversação em dupla:** participe conforme sua disponibilidade e pratique com outros alunos.
- **Aulas individuais**: sessões personalizadas com um professor para focar nos seus desafios.
- **Coaching para idiomas + gestão de projetos**: estratégias para otimizar seus estudos e manter a motivação.
- **Material complementar:** vídeos e conteúdos exclusivos em uma plataforma intuitiva.

Conheça o curso Fluência Descomplicada

Compartilhe

Ajude a democratizar o ensino do inglês no Brasil

Caro leitor,

Você chegou ao final de mais uma etapa importante em sua jornada de aprendizado de inglês.

Mas lembre-se, o conhecimento é ainda mais poderoso quando compartilhado.

Nossa missão vai além do ensino; é também sobre criar uma comunidade onde o inglês é acessível a todos, independentemente de onde estão ou de suas condições financeiras.

Ao compartilhar este material, você se torna parte fundamental dessa missão.

Como você pode ajudar?

Baixe este material

Faça o download deste material através do link fornecido para obter uma cópia e explorar seu conteúdo.

Compartilhe com alguém

Pode ser alguém próximo a você - amigo, colega, familiar ou alguém em sua rede social - que se beneficiaria deste aprendizado.

Espalhe a palavra

Converse sobre como este material ajudou você e encoraje outros a se juntarem a nós nesta jornada de aprendizado.

Cada pessoa que você ajuda a aprender inglês é uma semente plantada para um futuro onde barreiras linguísticas são diminuídas e oportunidades são ampliadas.

Juntos, podemos fazer uma grande diferença.

Vamos juntos nessa missão?

Vânia Paula de Freitas

Aula individual | Conversação | Inglês Corporativo Descomplicado