

COMMUNICATIVE APPROACH TO TEACHING ENGLISH FOR SOFTWARE DEVELOPMENT AND CLOUD COMPUTING SPECIALISTS

Iryna Kalynovska

PhD, Associate Professor,

Lesya Ukrainka Volyn National University, Ukraine

In today’s digital transformation era, where cloud computing and software development drive technological innovation, proficiency in English has become important for IT specialists in these fields. Knowledge of English is particularly vital for software developers and cloud engineers who need to read and understand technical documentation, program codes and API specifications, access the latest software development research and cloud computing updates, communicate in distributed development teams, and participate in international technical conferences, code reviews, and online development forums.

Traditional methods of learning English, which are often based on memorizing grammar rules and mechanical word learning, prove ineffective for software development and cloud computing professionals. These specialists require not only theoretical knowledge but also practical communication skills in real development situations.

The communicative approach to learning English offers an alternative teaching methodology based on “learning through communication”, whereby learners actively use the language in authentic contexts that align with their professional needs (Koliada & Kalynovska, 2023). This approach facilitates engaging and productive learning experiences, ensuring effective preparation of IT specialists.

The aim of our research is to examine the implementation of the communicative approach in teaching English to software developers and cloud computing specialists, with particular focus on technical communication and development practices.

The research material is the textbook *Professional English in Use ICT* by Cambridge University Press (Esteras & Fabre, 2007). The textbook covers essential topics in modern software development, including: text information processing, database management, multimedia applications, web development, and internet security protocols. The textbook is structured as a comprehensive learning resource that incorporates Information Communications Technology (ICT) terminology and concepts, making it particularly relevant for technical specialists.

The main principle of the communicative approach is that software developers and cloud engineers must not only acquire knowledge (e.g., know grammar, lexical and pronunciation norms) but also develop skills to use this knowledge for real communication purposes (Koliada & Kalynovska, 2023).

For teaching English in software development contexts, the textbook (Esteras & Fabre, 2007) offers exercises for mastering key language components:

- morphemes, focusing on software terminology, for example: *Correct each sentence using the words in brackets and the correct suffix from A opposite* (Esteras & Fabre, 2007, p. 79);

- specialized vocabulary related to software development: *Complete the sentences below with words from the box [repository, branch, pull request, commit, merge]* (Esteras & Fabre, 2007, p. 39); *Complete this article about the REST API architecture with words from the box [endpoint, request, response, authentication]* (Esteras & Fabre, 2007, p. 45);

- technical phrases common in development environments: *Look at B opposite and label this diagram with the correct terms* (Esteras & Fabre, 2007, p. 15); *Look at B, C opposite and underline all the classifying expressions in the documentation of microservices architecture* (Esteras & Fabre, 2007, p. 85);

- complex sentences: *Read A and B opposite and decide if these sentences are True or False. If they are false, correct them* (Esteras & Fabre, 2007, p. 53); *Draw lines between the columns to match development tools with their primary functions* (Esteras & Fabre, 2007, p. 47).

The practical application of the communicative approach in software development contexts is demonstrated through tasks such as: *Make a list of essential development tools and describe their role in your workflow* (Esteras & Fabre, 2007, p. 71); *Write a list of advantages and disadvantages of using microservices architecture* (Esteras & Fabre, 2007, p. 51); *Simulate a code review discussion using the provided technical vocabulary* (Esteras & Fabre, 2007, p. 53); *How have computers changed the way you work or study?* (Esteras & Fabre, 2007, p. 47).

The analysis of exercise instructions demonstrates how the communicative approach is systematically implemented through carefully structured instruction patterns (Esteras & Fabre, 2007). The instructions employ direct engagement techniques that progress from guided to autonomous communication: starting with controlled practice (*Complete the sentences below with words from the box*, p. 39) and advancing to open-ended professional communication scenarios (*How have computers changed the way you work or study?*, p. 47). This progression particularly benefits software developers, as it mirrors their professional growth from basic technical documentation comprehension to complex team interactions.

The source material’s instruction style strategically builds communicative competence through three key levels: individual technical vocabulary mastery (*Complete this article about the REST API architecture with words from the box*, p. 45), pair-based dialogue practice (*Read A and B opposite and decide if these sentences are True or False*, p. 53), and group-oriented professional discussions.

Such a layered approach effectively prepares developers for real-world communication scenarios they encounter in their work, from code documentation to team collaborations and client interactions. The integrated nature of the instructions, with frequent references to supplementary materials (*Look at B opposite*, p. 15; *Look at B, C opposite*, p. 85), supports the communicative approach’s emphasis on contextual learning, where language acquisition occurs within authentic professional contexts rather than in isolation.

In conclusion, the communicative approach to teaching English for software development and cloud computing specialists offers significant advantages: increased technical vocabulary, improved documentation skills, enhanced collaboration abilities, and better problem-solving communication. Communicative tasks develop essential skills for analysing technical information, making architectural decisions, articulating development solutions, and finding consensus in team discussions. These competencies are crucial for effective performance in modern software development and cloud computing environments.

References

Koliada, E. K., Kalynovska, I. M. (2023). Komunikatyvnyi pidkhid do vyvchennia inozemnoi movy v konteksti zabezpechennia yakosti vyshchoi osvity [The Communicative Approach to Foreign Language Learning in the Context of Higher Education Quality Assurance]. *Naukovi zapysky Natsionalnoho universytetu “Ostrozka akademiia”. Seriia “Filolohiia”* [Scientific Periodical *Scientific Proceedings of Ostroh Academy National University: Philology Series*]. № 17(85). P. 14–18. [http://doi.org/10.25264/2519-2558-2023-17\(85\)-14-18](http://doi.org/10.25264/2519-2558-2023-17(85)-14-18) [in Ukrainian].

Esteras, S. R., Fabre, E. M. (2007). Professional English in Use ICT. Retrieved from https://www.academia.edu/37663296/Professional_English_in_Use_ICT?email_work_card=title