DEVELOPING IT STUDENTS' ENGLISH SPEAKING AND COLLABORATIVE PROBLEM SOLVING SKILLS THROUGH CLIL

РАЗВИТИЕ НАВЫКОВ УСТНОЙ РЕЧИ НА АНГЛИЙСКОМ ЯЗЫКЕ И СОВМЕСТНОГО РЕШЕНИЯ ПРОБЛЕМ У СТУДЕНТОВ ИТ-НАПРАВЛЕНИЙ НА ОСНОВЕ ПОДХОДА CLIL

CLIL YONDASHUVI ASOSIDA IT YO'NALISHI TALABALARINING INGLIZ TILIDA OG'ZAKI NUTQ VA HAMKORLIKDA MUAMMOLARNI HAL QILISH KO'NIKMALARINI RIVOJLANTIRISH

Ikromova Muazzam Dilmurod kizi
Namangan State University, Faculty of World Languages, Associate professor of the Department of English and literature, PhD
Email:Dr.muazzam.dilmuradovna@gmail.com
Orcid: 0009-0002-4869-1915
DOI:https://doi.org/10.5281/zenodo.23741586

Abstract

This article proposes a CLIL framework for developing English speaking and collaborative problem solving through small software development projects in undergraduate IT education. A focused narrative synthesis connects integrated learning, task-based instruction, interaction research and professional computing requirements. The analysis produces a project structure in which technical decisions require students to clarify requirements, explain alternatives, negotiate priorities and respond to feedback in English. An illustrative equipment-booking application demonstrates how these communicative actions can be distributed across project stages. The proposed assessment separates individual speaking, interactional contribution and technical reasoning from the quality of the group product. Role rotation, information gaps and individual follow-up questions are used to make participation visible. The results are conceptual and methodological; no intervention or measured learning gains are reported. The article argues that software projects can create sustained purposes for English use when interaction is deliberately designed and supported. It also identifies unequal participation, excessive programming demands and rehearsed presentations as threats to valid assessment, and outlines a research design for evaluating the proposed approach.

Keywords: CLIL; IT students; English speaking; collaborative problem solving; software projects; interaction; professional communication.

Аннотация

В статье предлагается методическая система, основанная на подходе CLIL и направленная на развитие навыков устной речи на английском языке и совместного решения проблем у студентов ИТ-направлений бакалавриата посредством небольших проектов по разработке программного обеспечения. Целенаправленный аналитический обзор литературы объединяет

© Ushbu material Creative Commons Attribution 4.0 International (CC BY 4.0) litsenziyasi asosida tarqatiladi.

положения предметно-языкового интегрированного обучения, обучения на основе заданий, исследований речевого взаимодействия и требований к профессиональной подготовке специалистов в области информационных технологий. В результате анализа разработана структура проектной работы, в рамках которой принятие технических решений требует от студентов уточнения требований, объяснения альтернативных вариантов, согласования приоритетов и реагирования на обратную связь на английском языке. На примере приложения для бронирования оборудования показано, как указанные коммуникативные действия могут быть распределены по этапам проекта. Предлагаемая система оценивания предусматривает раздельную оценку индивидуальных навыков устной речи, вклада во взаимодействие и обоснованности технических решений, а также качества группового продукта. Для выявления участия каждого студента используются смена ролей, задания с информационным дефицитом и индивидуальные уточняющие вопросы. Результаты исследования имеют концептуальный и методический характер; педагогический эксперимент не проводился, количественные показатели прироста учебных достижений не представлены. Обосновывается положение о том, что проекты по разработке программного обеспечения могут обеспечивать устойчивую потребность в использовании английского языка при условии целенаправленного проектирования и поддержки взаимодействия. Кроме того, неравномерное участие студентов, чрезмерная сложность задач по программированию и заранее отрепетированные презентации рассматриваются как факторы, снижающие валидность оценивания. Намечен дизайн дальнейшего исследования для проверки эффективности предлагаемого подхода.

Ключевые слова: CLIL; студенты ИТ-направлений; устная речь на английском языке; совместное решение проблем; проекты по разработке программного обеспечения; взаимодействие; профессиональная коммуникация.

Annotatsiya

Ushbu maqolada bakalavriatning IT yo‘nalishi talabalarida dasturiy ta’minot ishlab chiqishga oid kichik loyihalar orqali ingliz tilida og‘zaki nutq va hamkorlikda muammolarni hal qilish ko‘nikmalarini rivojlantirishga qaratilgan CLIL asosidagi metodik tizim taklif etiladi. Adabiyotlarning maqsadli tahliliy sharhi predmet va tilni integratsiyalashgan holda o‘qitish, topshiriqlarga asoslangan ta’lim, nutqiy o‘zaro ta’sirga oid tadqiqotlar hamda axborot texnologiyalari mutaxassislarini kasbiy tayyorlash talablarini o‘zaro bog‘laydi. Tahlil natijasida texnik qarorlar qabul qilish jarayonida talabalardan talablarni aniqlashtirish, muqobil yechimlarni izohlash, ustuvor vazifalarni kelishib olish va fikr-mulohazalarga ingliz tilida munosabat bildirishni talab etadigan loyiha tuzilmasi ishlab chiqilgan. Jihozlarni band qilish ilovasi misolida ushbu kommunikativ harakatlarni loyiha bosqichlari bo‘yicha taqsimlash imkoniyatlari ko‘rsatilgan. Taklif etilayotgan baholash tizimida talabaning individual og‘zaki nutqi, o‘zaro muloqotga qo‘shgan hissasi va texnik qarorlarni asoslash qobiliyati guruh tomonidan yaratilgan mahsulot sifatidan alohida baholanadi. Har bir talabaning ishtirokini aniqlash uchun rollarni almashtirish, axborot yetishmovchiligiga asoslangan topshiriqlar va individual aniqlashtiruvchi savollardan foydalaniladi. Tadqiqot natijalari konseptual va metodik xususiyatga ega bo‘lib, pedagogik tajriba-sinov o‘tkazilmagan hamda o‘quv natijalaridagi o‘sishning miqdoriy ko‘rsatkichlari keltirilmagan. Maqolada o‘zaro muloqot maqsadli loyihalashtirilgan va tegishli tayanchlar bilan ta’minlangan sharoitda dasturiy loyihalar ingliz tilidan foydalanishga muntazam ehtiyoj yaratishi

© Ushbu material Creative Commons Attribution 4.0 International (CC BY 4.0) litsenziyasi asosida tarqatiladi.

mumkinligi asoslanadi. Shuningdek, talabalar ishtirokining notekisligi, dasturlash vazifalarining haddan tashqari murakkabligi va oldindan mashq qilingan taqdimotlar baholash validligini pasaytiruvchi omillar sifatida ko'rib chiqiladi hamda taklif etilgan yondashuv samaradorligini tekshirish uchun keyingi tadqiqot loyihasi belgilab beriladi.

Kalit so'zlar: CLIL; IT yo'nalishi talabalari; ingliz tilida og'zaki nutq; hamkorlikda muammolarni hal qilish; dasturiy ta'minot ishlab chiqish loyihalari; o'zaro muloqot; kasbiy kommunikatsiya.

Introduction

A software project requires decisions that other people must understand. Even a small student application involves questions about user needs, priorities and the reasons for choosing one solution over another. Computing Curricula 2020 recognises communication and teamwork as part of professional preparation [1]. English lessons can use these decisions as occasions for purposeful interaction.

The instructional challenge is that completing a group project does not necessarily give every student meaningful speaking practice. One learner may do most of the coding, another may prepare the slides and a confident speaker may deliver the final presentation. The group can produce a plausible application while some members remain unable to explain its purpose or answer an unexpected question in English. A CLIL project therefore needs a planned distribution of communicative responsibility.

CLIL requires attention to both content and language learning [2]. In the framework developed here, the content includes identifying requirements, comparing simple design alternatives and explaining how a feature addresses a user need. The language includes asking for clarification, giving reasons, expressing disagreement and revising an explanation. The project supplies continuity across these actions, while individual tasks make each student's contribution observable.

The CEFR Companion Volume includes mediation and online interaction alongside other language activities [3]. These categories are relevant to a team that must interpret a user request or resolve different understandings of a requirement. They do not, however, establish that any collaborative software project will improve English. Learning opportunities depend on what students are required to say, how others respond and what feedback follows.

This article asks how a small software project can generate sustained English interaction, how teachers can support participation without scripting every exchange, and how assessment can separate individual development from group achievement. Its contribution is a proposed project structure and assessment plan. The intended context is a university English course coordinated with introductory IT teaching, rather than a professional software engineering course taught entirely in English.

Literature Review

The CLIL literature provides a basis for connecting subject content, communication and thinking [2]. In this proposal, a technical choice creates the subject matter of an exchange, while the need to justify that choice creates the language demand. The cultural dimension concerns responsible participation, audience awareness and respectful negotiation. It should not be reduced to assumptions about how members of different national groups communicate.

Ellis distinguishes meaning-focused tasks from exercises organised primarily around practising specified language forms [4]. This distinction helps identify whether a project meeting has a communicative purpose. If students merely read predetermined sentences, the activity offers limited

© Ushbu material Creative Commons Attribution 4.0 International (CC BY 4.0) litsenziyasi asosida tarqatiladi.

evidence of problem solving. If they must resolve a difference in requirements and reach a defensible decision, the exchange has an outcome that depends on meaning.

Storch's study of adult ESL pair work identifies different interaction patterns and shows that pairing learners does not automatically produce the same learning opportunities [5]. Its relevance here is a caution about participation, not direct evidence for IT projects. The proposed design therefore includes role rotation and individual responsibility, while recognising that these procedures must be examined rather than assumed to create productive collaboration.

The Scrum Guide describes an iterative framework for work involving inspection and adaptation [6]. The present article borrows the broad idea of returning to a product and revising it after feedback. It does not claim that the classroom sequence implements Scrum faithfully or that teaching simplified project meetings develops professional Scrum competence. The educational purpose is sustained communication around a changing technical problem.

Black and Wiliam connect formative assessment with decisions that influence subsequent learning [7]. In the proposed project, feedback is used before the final performance so that students can change how they explain, question or justify. The important instructional event is the revised performance. Collecting extensive comments without giving learners time to respond would not accomplish the same purpose.

Bruton's critique of CLIL research warns against attributing group differences to the instructional label without considering selection and exposure [8]. A future evaluation of project-based CLIL should therefore account for prior English proficiency, programming experience, contact time and differences in teacher support. Enthusiastic presentations and positive student perceptions would be insufficient evidence of causal effectiveness.

Together, these sources support a design that treats collaboration as an activity to be structured and observed. A software product provides shared material for discussion, but language development requires more than access to that material. The following sections specify the decisions, speaking opportunities and evidence that could make the project educationally useful. These specifications are author-developed methodological results, not summaries of an already completed intervention.

Materials and Methods

The study uses a focused narrative synthesis and conceptual project design. Eight sources were selected for their relevance to computing competence, integrated learning, language activity, task design, interaction and assessment. Bibliographic information and accessible source descriptions were checked through official, institutional or publisher resources. The selection was purposive and does not constitute a systematic review of all studies on project-based CLIL or English for IT students.

The analytical procedure begins with a professional communicative action, such as clarifying an ambiguous requirement. It then identifies a simple technical situation that requires that action, specifies the language support likely to be useful and determines what evidence would show successful participation. A final check asks whether the evidence belongs to the individual or only to the team. This check is necessary because the central educational objective includes each learner's speaking development.

The constructed project is an equipment-booking application for a university setting. It is deliberately narrow: users need to view equipment, request a booking and understand whether a request is accepted. The classroom product may be a paper prototype, a clickable mock-up or a small

© Ushbu material Creative Commons Attribution 4.0 International (CC BY 4.0) litsenziyasi asosida tarqatiladi.

implementation, depending on students' prior knowledge. No production deployment, real personal records or institutional system access is required by the design.

The project is organised across four teaching sessions with additional preparation between them. The duration is illustrative rather than empirically determined. Students work in small teams and rotate communicative responsibilities, including requirements analyst, developer, tester and user representative. These classroom responsibilities are not presented as a complete description of professional software roles.

Three design criteria guide the results. Every session must include a decision that can be explained in English. Every learner must contribute to an interaction in which a partner's response matters. Every assessment claim about individual ability must rest on individual evidence. A task failing one of these criteria is revised by changing the information distribution, the expected product or the follow-up questioning.

The study has no recruited participants, recorded meetings or measured outcomes. All project events and utterances are illustrative. The results are the proposed sequence, a communication matrix and an assessment protocol. Their feasibility and effectiveness require subsequent classroom piloting with appropriate participant information, consent for research recordings and transparent reporting of the comparison condition.

Results

Designing decisions that require communication

The first result is a project structure built around four decision points. Students first clarify the user's request, then compare a small set of design alternatives, examine a test result and revise their proposed solution. Each decision produces an artefact and an English explanation. The continuity of the equipment-booking problem allows vocabulary and communicative functions to recur without making every exchange identical.

At the requirements stage, the user representative receives constraints that the developer does not yet know. For example, some equipment may be unavailable for part of the day, and a user may need more than one item. The developer must ask questions before proposing a booking rule. The task succeeds when both participants can state the agreed requirement in consistent terms. A completed form without an exchange would not meet the speaking objective.

At the design stage, students compare two simple ways of presenting availability. The alternatives should each have an understandable advantage and limitation. Learners must explain how their choice affects the user, rather than declare one interface "better" without grounds. A partner's challenge asks them to reconsider an assumption or provide an example. The task therefore links technical evaluation with reason-giving and responsive speech.

At the testing stage, a scenario exposes a problem such as two requests for the same item at the same time. Students explain the observed behaviour, distinguish it from the expected behaviour and propose what should change. The technical complexity remains limited to the concepts already taught. The English teacher evaluates the clarity of the explanation while consulting a computing teacher about the plausibility of the proposed solution.

At the revision stage, the team explains what changed and why. The final presentation is followed by individual questions about a requirement, a decision or a limitation. Students may refer to the prototype, but each must formulate a response. The project thus includes prepared communication and a bounded opportunity for spontaneous explanation.

© Ushbu material Creative Commons Attribution 4.0 International (CC BY 4.0) litsenziyasi asosida tarqatiladi.

This structure creates a plausible advantage through repeated, consequential use of English. A clarification can alter the requirement; an explanation can persuade a teammate; a test discussion can change the product. These consequences give communicative actions a purpose within the learning task. Whether they lead to improved speaking must still be established through individual performance evidence across time.

Mapping project stages to individual evidence

The second result is a planning matrix. It links each stage with a language function and an individual evidence source. The product column makes the technical outcome visible, while the final column prevents the group artefact from standing in for every learner's competence.

Stage	English speaking function	Team product	Individual evidence
Requirements	Ask and confirm meaning	Agreed requirement note	Clarification exchange
Design	Compare and justify	Annotated prototype	Explanation of one choice
Testing	Describe and diagnose	Issue report	Expected versus actual account
Revision	Respond and qualify	Revised prototype	Answer to an unfamiliar question

The teacher can use this matrix to distribute participation. A learner who explained the design in one session should ask questions or report a test in another. Rotation broadens opportunities but does not require identical amounts of speech in every exchange. Some contributions are brief yet important, such as identifying an ambiguous requirement or asking a question that changes the team's decision.

Information gaps should remain meaningful. Giving different students arbitrary fragments of a text can force talk without creating technical reasoning. In the proposed project, the missing information concerns a user constraint, a design assumption or a test outcome. Students need that information to reach the agreed decision. The resulting exchange can therefore be assessed for both relevance and communicative effectiveness.

Language support may include a short bank of expressions for clarifying conditions, comparing alternatives and acknowledging a limitation. The teacher should model how these expressions serve a purpose, then vary the problem so that students must choose their own content. A phrase bank becomes less useful when learners reproduce it mechanically without responding to what a partner has said.

The matrix can also support asynchronous preparation, but the central speaking evidence should remain audible and attributable to the learner. A polished written contribution or a slide prepared by a teammate cannot substitute for the individual oral action specified in the final column.

Assessment and feedback during the project

The third result is an assessment protocol with separate records for speaking, interaction and technical reasoning. Speaking concerns intelligibility, organisation and appropriate language for the purpose. Interaction concerns relevant responses, clarification and contribution to joint understanding.

© Ushbu material Creative Commons Attribution 4.0 International (CC BY 4.0) litsenziyasi asosida tarqatiladi.

Technical reasoning concerns whether the explanation fits the specified requirement and evidence. The group product is assessed separately so that a technically strong team cannot conceal an individual's limited communicative participation.

A proposed five-point scale can describe each individual dimension. At the lower end, the learner cannot communicate the relevant meaning despite available support. At the middle, the learner conveys a generally understandable contribution but needs prompts or leaves an important relation unclear. At the upper end, the learner explains accurately, responds appropriately and adjusts to the partner's question under the stated conditions. Intermediate points require examples from locally collected performances before formal use.

The teacher should use short evidence notes rather than attempt to transcribe every meeting. A note can identify the task, the learner's action and its consequence: for example, the student asked whether bookings could overlap and the team revised its rule. Such an observation provides more useful feedback than a general statement that the learner was active. It also avoids equating participation with the number of words spoken.

Feedback should occur while the project is still developing. The teacher may select one recurring language issue and one interaction issue for focused attention. Students then repeat a comparable exchange with a changed constraint. This allows the teacher to observe whether the feedback affects the next performance. Correcting every grammatical error during a decision-making discussion may interrupt the exchange and obscure its purpose.

Self-assessment asks students to identify a moment when their question or explanation influenced the team. They should support the claim with a decision note or a recalled exchange and identify what they would change next time. Peer feedback can comment on clarity and responsiveness, but it should not become the sole basis for an individual grade. Friendships, confidence and uneven technical knowledge may affect peer judgements.

At the end, the teacher combines evidence from different stages rather than relying only on the final presentation. A brief individual transfer task introduces a related but unfamiliar requirement. The learner explains how it would affect the proposed design. This final task is especially useful for distinguishing flexible understanding from a rehearsed account of the team's completed work.

Discussion

Software projects can give English speaking a sustained purpose because decisions remain connected across lessons. Students return to a requirement, defend a choice and reconsider it after testing. This continuity allows the same communicative function to appear in different situations. A learner who first asks for clarification about availability can later use the function to question a test result. The proposed benefit is repeated practice with changing meaning, rather than repeated delivery of an identical script.

The framework also makes technical reasoning available for discussion. A prototype or issue report gives students something specific to point to, explain and question. This can reduce the burden of inventing content for a general speaking task. It does not remove linguistic difficulty, however. Learners still need support for expressing conditions, uncertainty and disagreement. Teachers should diagnose which functions prevent participation rather than assume that technical interest alone will produce fluent English.

Unequal participation is a central risk. A technically experienced student may control decisions even when another student speaks more English. Conversely, a confident English speaker may present a

© Ushbu material Creative Commons Attribution 4.0 International (CC BY 4.0) litsenziyasi asosida tarqatiladi.

choice that others developed. The proposed individual evidence records address this problem by asking what each learner understood and contributed. They cannot fully reveal every informal exchange, but they provide a more defensible basis than assigning the same communication score to the whole team.

Programming demands can also overwhelm the language objective. If students spend most of the lesson fixing unfamiliar technical problems, there may be little time for the planned interaction. A paper or clickable prototype can preserve requirements analysis and design discussion while reducing implementation load. The choice of product should follow the content objective and students' preparation. A more complex application is not automatically a better CLIL task.

Teachers should also distinguish productive hesitation from disengagement. A learner may pause to formulate an accurate explanation or reconsider a technical assumption. Immediate correction or replacement by a more fluent peer can remove the opportunity to complete that reasoning. Providing preparation time and allowing reference to a diagram can support participation without supplying the entire answer.

The framework is compatible with multilingual preparation when its role is explicit. Students may clarify a concept in a familiar language and then produce the agreed English explanation. The teacher should record major support conditions when assessing independence. The desired balance depends on the task stage; the final evidence must still show the English communicative action that the lesson set out to develop.

Limitations and future evaluation

This article presents a conceptual framework rather than evidence of effectiveness. The selected sources come from related but different settings, and their implications cannot be transferred automatically to undergraduate IT students in Uzbekistan or elsewhere. The equipment-booking project is deliberately narrow. Other domains may require different prerequisites, discourse functions and assessment criteria.

A future evaluation should compare the proposed sequence with a clearly described alternative offering similar teaching time and technical content. Baseline assessment should include both speaking and relevant computing knowledge. Researchers should collect individual outcomes, because improvements in the group product may reflect only one member's expertise. Where possible, assessors should judge anonymised recordings using calibrated criteria and report their level of agreement.

Interaction analysis can supplement outcome scores by identifying how decisions are negotiated. Relevant observations include clarification requests, responses to challenge and revisions after feedback. Frequency alone is insufficient: a repeated question may reflect confusion, while a single well-placed question may resolve the problem. Qualitative evidence should therefore connect a communicative action with its function and consequence in the project.

Retention and transfer require additional tasks after the project ends. A new user constraint or a different simple application can test whether learners reuse the communicative functions outside the rehearsed scenario. Research should also document workload, absences, role distribution and access to technical support. These factors can explain why an otherwise coherent design succeeds in one class and proves difficult in another.

Conclusion

CLIL-based software projects can organise English speaking around meaningful technical decisions. The proposed framework connects requirements clarification, design comparison, testing and revision with specific language functions and individual evidence. Its main methodological contribution

© Ushbu material Creative Commons Attribution 4.0 International (CC BY 4.0) litsenziyasi asosida tarqatiladi.

is the separation of speaking development and interactional contribution from the quality of the final group product.

The potential advantages depend on purposeful information gaps, manageable content, role rotation and opportunities to act on feedback. A project should give every learner a reason to explain, question and respond, while preserving a clear technical learning objective. The framework offers a practical basis for a small classroom pilot and a transparent evaluation. Its effectiveness should be judged through individual performance, transfer and the quality of collaboration, rather than through the attractiveness of the completed application alone.

References

1. Association for Computing Machinery, IEEE Computer Society. Computing Curricula 2020: Paradigms for Global Computing Education. – New York: ACM, 2020. – URL: https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2020.pdf
2. Coyle D., Hood P., Marsh D. CLIL: Content and Language Integrated Learning. – Cambridge: Cambridge University Press, 2010. – 184 p. – DOI: 10.1017/9781009024549.
3. Council of Europe. Common European Framework of Reference for Languages: Learning, Teaching, Assessment – Companion Volume. – Strasbourg: Council of Europe Publishing, 2020. – URL: https://www.coe.int/en/web/common-european-framework-reference-languages/cefr-companion-volume-and-its-language-versions.
4. Ellis R. Task-based language teaching: Sorting out the misunderstandings // International Journal of Applied Linguistics. – 2009. – Vol. 19, no. 3. – P. 221-246. – DOI: 10.1111/j.1473-4192.2009.00231.x.
5. Storch N. Patterns of interaction in ESL pair work // Language Learning. – 2002. – Vol. 52, no. 1. – P. 119-158. – DOI: 10.1111/1467-9922.00179.
6. Schwaber K., Sutherland J. The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. – 2020. – URL: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf.
7. Black P., Wiliam D. Developing the theory of formative assessment // Educational Assessment, Evaluation and Accountability. – 2009. – Vol. 21, no. 1. – P. 5–31. – DOI: 10.1007/s11092-008-9068-5.
8. Bruton A. Is CLIL so beneficial, or just selective? Re-evaluating some of the research // System. – 2011. – Vol. 39, no. 4. – P. 523–532. – DOI: 10.1016/j.system.2011.08.002.

© Ushbu material Creative Commons Attribution 4.0 International (CC BY 4.0) litsenziyasi asosida tarqatiladi.