- **Course Title:** English for Software Professionals **(Student Success Introductions)**
- **CEFR Level:** B2
- **Lesson Number:** 1
- **Topic:** Identifying Learning Styles and Strategies
- **Lesson Duration:** 3 hours (1hr20 - break 20mins - 1hr20)
- **Can-Do Objectives:** (Aligned with CEFR descriptors)
    - I can evaluate different learning approaches and select strategies most suited to my needs.
    - I can assess my learning style and adapt strategies accordingly.
    - I can set realistic language learning goals to support professional development.

Materials

- Handouts:
    - Strategy Reflection
    - Weekly Planner Template

- Audio/Video Files:
    - Short video: “What’s Your Learning Style?” (Visual, Auditory, and Kinesthetic Learning Styles)

- Required Tech / Supplies:
    - Projector

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
learning style	An individual’s preferred way of acquiring and processing information	My learning style is visual—I remember things best with images.	N
strategy	A plan of action designed to achieve a specific goal	Using flashcards is one strategy I use to memorize vocabulary.	Y
reflect	To think carefully about something	It’s important to reflect on your progress every few weeks.	Y
goal-setting	The process of identifying something you want to achieve and planning how	Effective goal-setting helps improve your motivation.	Y
adaptability	The ability to change your approach based on new conditions	Adaptability is essential when learning a new skill.	Y

Lesson Structure (PPP)

- **Warm-Up / Review: Introduction to Learning Styles and Strategies (10–15 mins)**
    - Step 1: Context hook (3–5 mins)
        - Begin with a short professional scenario:
          "Imagine your company offers a new online English training program. Some employees prefer watching videos, others like doing quizzes, and some want live discussions. Which approach would you choose?"
        - Ask a few volunteers to explain their choice and why it fits their learning preferences.
    - Step 2: Brainstorm (5 mins)
        - On the board, write three columns labeled: *Visual*, *Auditory*, *Kinesthetic / Practical.*
        - Elicit examples of workplace learning activities that fit each style:
            - Visual: "Reading slides or infographics."
            - Auditory: "Listening to podcasts or webinars."
            - Kinesthetic: "Practicing role-plays or using real tools."
                - Add a few of your own to balance ideas.
    - Step 3: Pair discussion (5 mins)
        - Students discuss in pairs: "How do you prefer to learn at work?," "What strategies help you remember or apply new skills?"
        - Ask one or two pairs to share their answers with the class.
    - Step 4: Transition
        - Summarize: "In this lesson, we'll look at how identifying your learning style can help you choose effective strategies and set clear, realistic goals for your professional English development."
    - **Materials:** Whiteboard and markers

I. Presentation (30–40 mins)

- **Vocabulary Introduction (15 mins)** – Students are introduced to five core vocabulary terms related to learning and strategy.
    - Step 1: Write all five terms on the board: learning style, strategy, reflect, goal-setting, adaptability.
        - Ask: "Which of these words do you already use in your work or study life?"
        - Additional teacher prompt: "In your last training, did you use goal setting? Have you ever had to be adaptable at work?"
    - Step 2: Pair-share: Students choose one word they know well, then explain it with a personal example.

- **Grammar/Function Focus (10 mins)** – Focus on phrases for describing habits and preferences (e.g., "I tend to...", "I usually...", "I'm more of a...").
    - Step 1: Write structures on board.

    - Step 2: Model examples: "I tend to remember things better when I draw them."
    - Step 3: Students orally create and write 2 sentences about their study habits.
    - Step 4: Share with a partner.

- **Mini-Lecture & Guided Discussion (10–15 mins)** – Teacher introduces the concept of learning styles and facilitates discussion.
    - Step 1: Introduce 3 learning styles orally: Visual, Auditory, Kinesthetic.
        - Ask guiding questions:
            - "Which type do you think you are?"
            - "Can you be more than one?"
    - Step 2: Play short video: What's Your Learning Style? (Visual, Auditory, and Kinesthetic Learning Styles)
    - Step 3: Class Discussion: "Which ideas from the video did you connect with?"
    - Step 4: Students critique the learning styles
        - Teacher writes critique sentence stems on the board:
            - "This model is useful when…but weak when…," "A limitation is…," "On balance…"
            - 1-2 learners share aloud using hedging language
    - **Materials:** Whiteboard and markers, projector

II. Practice (30–40 mins)

- **Controlled Activities (20 mins)** – Strategy Reflection
    - Step 1: Students complete a checklist of strategies that match their learning style.
    - Step 2: Students list two concrete behaviors tied to their style:
        - Students share with their partner "I'll know it helps if…."
    - Step 3: Match 3 strategies to business-related language tasks (e.g., presentations, writing emails).
    - Step 4: Peer-pair discussion: "Which strategy would help you most with English at work?"
    - **Materials:** Strategy Reflection Handout

- **Reflection (10–15 mins)**
    - Step 1:
        - Oral journal prompt:
            - "What is one new strategy you will try this week and why?"
            - Volunteers share aloud.
    - Teacher prompts discussion: "Why is it important to reflect on your learning habits regularly?"

[20-Minute Break]

III. Production (30–40 mins)

- **Scenario-Based Planning Task (30–40 mins)** – Create a 1-month language study plan for a professional assignment.
    - Step 1: Present scenario: "You are preparing for an English-speaking work project in one month."
    - Step 2: Students form small groups and use a weekly planner to outline their strategy-based study plan. Students include:
        - time, method, and expected outcomes
    - Step 3: Language Bank Upgrade
        - Post a “Language Upgrade Bank” on the board and explain that students should integrate at least one expression from each section into their discussions and presentations.
            - Hedging: tend to, broadly, in many cases, arguably
            - Contrast: whereas, even though, however
            - Recommendation: I’d recommend, more effective than
        - Briefly model how to use them in a professional way:
            - “In many cases, group study is more effective than working alone.”
            - “I’d recommend focusing on presentation skills rather than grammar drills.”
            - “Visual learners tend to remember new terms faster.”
    - Step 4: Groups present and explain their plan.
    - Teacher prompts: "What would you do differently if you had more time?" "Which activity will be most helpful?"
- **Materials:** Weekly Planner Template

IV. Digital Tool (45 mins)

- To provide students with the opportunity to use the digital tool in class with teacher support.
- The teacher can demonstrate activities with the whole class and/or support students as they work individually, for example:
    - Provide a tutorial on the digital tool and its functions
    - Show students how to login
    - Try different activities with teacher support

V. Wrap-Up (15 mins)

- **Vocabulary Review (5–7 mins)**
    - Oral quiz: Call out definitions; students respond with the correct term.
    - Ask: “Can you use one new word in a sentence related to your work?”

- **Self-Reflection (8–10 mins)**
    - Step 1: Ask students to write bullet point answers to:
        - "What did I learn about myself today?"

        - "What will I try differently this week?"
    - Step 2: Pair-share and discuss, then 2-3 volunteers present their answers.
    - Teacher note: Focus on improving clarity and upgraded lexis..

Optional Independent Practice

- Choose one new learning strategy from today's lesson and try it twice this week.
- Keep a log or voice memo describing what you did and how it helped.

Notes for the Instructor

- Pacing: Adjust vocabulary and discussion time based on class engagement.
- Corrections: Focus on modeling correct usage rather than interrupting fluency.
- Differentiation: Allow choice in strategy tasks; visual learners may draw plans, auditory learners may explain aloud.
- Encourage: students to revisit strategies in future lessons and compare effectiveness.

- **Course Title:** English for Software Professionals **(Student Success Introductions)**

- **CEFR Level:** B2
- **Lesson Number:** 2
- **Topic:** Setting SMART Goals for Business English
- **Lesson Duration:** 3 hours (1hr20 - break 20mins - 1hr20)
- **Can-Do Objectives:** (Aligned with CEFR descriptors)
    - I can create SMART goals tailored to my professional needs.
    - I can monitor and adjust my goals over time.

Materials

- Handouts:
    - SMART Goal Baseline

- Audio/Video Files:
    - Sample video on goal setting in the workplace (How to Set SMART Goals: Goal Setting for Businesses)

- Required Tech / Supplies:
    - Whiteboard and markers
    - Projector

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
SMART Goal	A goal that is Specific, Measurable, Achievable, Relevant, and Time-bound	My SMART goal is to improve my email writing by the end of June.	Y
milestone	A significant stage or event in a process or project	Reaching 100 new clients is a major milestone in our campaign.	Y
progress	Forward or onward movement toward a goal	I track my progress weekly to stay motivated.	Y
accountability	The quality of being responsible for your actions or decisions	Having a study partner helps with accountability.	Y
prioritize	To arrange or deal with in order of importance	I prioritize urgent tasks first each morning.	Y

Lesson Structure (PPP)

- **Warm-Up / Review: Review of Previous Lesson (10–15 mins)**
  Students revisit their identified learning strategies and share updates on how they tried them.
    - Step 1: Ask: "Which learning strategy have you tried since the last class?" List on the board.
    - Step 2: Pair up and describe what worked or didn't work.
    - Step 3: Distribute a quick reflection form to jot down insights (1–2 sentences).
    - Step 4: Teacher prompts: "Did your strategy help with your English at work this week?"
    - **Materials:** Reflection form previous lesson's learning strategies handout

I. Presentation (30–40 mins)

- **Vocabulary Introduction (15 mins)** – Students explore five goal-setting and planning terms.
    - Step 1: Write vocabulary terms on the board. Ask: "Have you seen these in a work or school setting?"
    - Step 2: Model pronunciation and usage. Students repeat and write personal examples.
    - Step 3: Pair-share to compare sentences.
    - **Materials:** Whiteboard and markers

- **Grammar/Function Focus (10 mins)** – Practice language for making and evaluating goals (e.g., "I plan to...", "My goal is to...", "I will track...").
    - Step 1: Write sample goal-setting phrases on the board.
    - Step 2: Ask students to build complete SMART goal sentences.
    - Step 3: Partner practice.
    - **Materials:** Whiteboard and markers

- **Mini-Lecture & Guided Discussion (10–15 mins)** – Explanation of SMART framework with real-life examples.
    - Step 1: Introduce SMART acronym orally: Specific, Measurable, Achievable, Relevant, Time-bound.
    - Step 2: Play short video: SMART Goals in Business (How to Set SMART Goals: Goal Setting for Businesses)
        - Discussion prompts:
            - "Which part of SMART is hardest for you?"
            - "Why is it important to set measurable goals?"
    - Step 3: SMART debate prompt
        - The teacher asks "Is SMART always ideal for language learning? What's missing?"
            - Students discuss in pairs or small groups, then do a quick class share

- Optional: add “E = Evaluate” and “R = Readjust” making SMART into SMARTER

II. Practice (30–40 mins)

- **Controlled Activities (20 mins)** – SMART Goal Baseline
  - Step 1: Students brainstorm a business English goal and apply the SMART framework to it.
  - Step 2: When students write their SMART goals, require a measurement line. For example:
    - “Baseline: I can write 120 words in 10 minutes. Target: 180 words in 10 minutes by end of month.”
    - Teacher tell students “Your goal must include both your current baseline and your target result.”
  - Step 2: Peer review goals using a checklist.
  - Step 3: Revise based on feedback.
  - **Materials:** SMART Goal Baseline Worksheet

- **Reflection (10–15 mins)**
  - Step 1: Students write about a personal or professional goal they’ve achieved and what helped.
  - Step 2: Teacher prompts: "How can you stay motivated over time?"

[20-Minute Break]

III. Production (30–40 mins)

- **Goal Coaching Role-Play (30–40 mins)** – Students practice advising one another on goal setting.
  - Step 1: Model and critique a full SMART goal
    - Teacher presents a full SMART goal with numbers and date on the board. Example:
      - “My SMART goal: I will increase my business email accuracy from 62% to 80% in six weeks by practicing twice a week with a peer.”
    - Student pairs critique using these questions: “What’s strong? What could we improve?”
  - Step 2: In pairs, one student is the “coach,” the other is the “learner.”
    - Role-play conversation prompts:
      - “What is your goal?”
      - “Why is it important?”
      - “How will you track your progress?”
    - Switch roles and repeat.
  - Step 3:
    - Class debrief: Discuss differences and lessons learned.

    - Role-play prompts can be written on the board:
        - “Describe your goal”
        - “Explain why it matters”
        - “How will you measure success?”

IV. Digital Tool (45 mins)

- To provide students the opportunity to use the digital tool in class with teacher support.
- The teacher can demonstrate activities with the whole class and/or support students as they work individually.

V.. Wrap-Up (15 mins)

- **Vocabulary Review (5–7 mins)**
    - Step 1: Write scrambled SMART goal terms on the board; students unscramble orally.
    - Ask: “Use one new word in a sentence about your work goal.”

- **Self-Reflection (8–10 mins)**
    - Step 1: Write response to: "What is one SMART goal you want to achieve by the end of this course?"
    - Step 2: Pair-share and record goals in notebook or on classroom board.

Optional Independent Practice

- Submit an update or brief video reflection before Lesson 4.

Notes for the Instructor

- Pacing tip: Allow students more time to revise goals if needed.
- Differentiation: Advanced learners can write multiple SMART goals; others can work from guided templates.
- Encourage accountability partners within the class to check in on each other’s progress.

- **Course Title:** English for Software Professionals **(Student Success Introductions)**
- **CEFR Level:** B2

- **Lesson Number:** 3
- **Topic:** Time Management and Prioritization
- **Lesson Duration:** 3 hours (1hr20 - break 20mins - 1hr20)
- **Can-Do Objectives:** (Aligned with CEFR descriptors)
    - I can apply time management techniques to my studies and work.
    - I can prioritize tasks based on urgency and importance.
    - I can describe how I organize and justify my schedule in professional settings.

Materials

- Handouts:
    - Role-Play: Team Planning Scenario
    - Eisenhower Matrix Template
    - Applying the Eisenhower Matrix

- Audio/Video Files:
    - Video: The Eisenhower matrix: How to manage your tasks with EISENHOWER

- Required Tech / Supplies:
    - Projector

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
prioritize	To arrange tasks in order of importance	I prioritize urgent emails before working on reports.	Y
deadline	A time or date by which something must be completed	My deadline for submitting the proposal is Friday at noon.	Y
schedule	A plan that gives expected times for activities or events	I follow a weekly schedule to manage meetings and tasks.	Y
procrastinate	To delay doing something that should be done	I sometimes procrastinate when a task feels overwhelming.	N
time-blocking	A productivity technique that divides your day into blocks for tasks	I use time-blocking to ensure I focus on one thing at a time.	Y

Lesson Structure (PPP)

- **Warm-Up: Review of SMART Goals (10–15 mins)**
  - Step 1: Ask students to retrieve their SMART Goal Baseline worksheets.
  - Step 2: In pairs, students review each other's goals and discuss:
    - "Was your goal realistic?"
    - "What time-based actions did you include?"
  - Step 3: Elicit from the class what challenges they might face in achieving their goals.
  - Step 4: Transition into today's topic by asking: "What do you need in order to meet your goals? How do you manage your time?"
  - **Materials:** SMART Goal Baseline handout from previous lesson

I. Presentation (30–40 mins)

- **Vocabulary Introduction (10 mins)** – Students learn five key vocabulary terms related to scheduling and prioritization.
  - Step 1: Write terms on the board.
  - Step 2: Provide definitions, example sentences, and ask students to give their own work-related examples.
  - Step 3: Ask students: "Which of these is most important for your success at work? Rank them from 1-5."
    - Students work in pairs to justify their answers: "I think deadline is #1 because…"

- **Grammar/Function Focus: Explaining Schedules and Justifying Priorities (10–15 mins)** – Students learn how to explain what they're doing and why.
  - Step 1: Write example: "I'm working on the client proposal first because it's due tomorrow."
  - Step 2: Highlight use of time references (e.g., "first," "after," "by Friday") and cause/effect connectors (e.g., "because," "so that").
  - Step 3: Practice orally: Students explain what they'll do this week and why, using these phrases.
  - Teacher prompt: "Tell your partner about 2 things on your to-do list this week. Why are they important?"

- **Mini-Lecture & Guided Discussion (10–15 mins)** – Students learn about different time management frameworks, focusing on the Eisenhower Matrix.
  - Step 1: Show video The Eisenhower matrix: How to manage your tasks with EISENHOWER

- Step 2: Distribute Eisenhower Matrix template. Teacher explains each quadrant with workplace examples.
- Step 3: Class discussion: “What types of tasks are usually urgent but not important?”
- Step 4: Lead into Practice section: “Let’s apply this to your own workday.”
- **Materials:** Eisenhower Matrix Template, projector

II. Practice (30–40 mins)

- **Controlled Activities (15 mins)** – Students apply prioritization to sample business tasks.
  - Step 1: Distribute the Applying Eisenhower Matrix worksheet with workplace-specific tasks (e.g., “Prepare monthly report,” “Answer client email,” “Update project tracker,” “Reply to project manager,”).
  - Step 2: In pairs, students place each task into their Eisenhower Matrix Template.
  - Step 3: Ask students to compare choices with another group and justify placement.
  - Teacher prompt: “Why did you put that in ‘Urgent/Not Important’?”
    - Emphasize: “How would this play out at work?”
  - **Materials:** Applying Eisenhower Matrix handout, Eisenhower Matrix Template

- **Deeper Debate on Prioritization: Justify & Anticipate Consequences (5 mins)**
  - Teacher prompts: “Why did you put this task in Urgent/Not Important?,” “What happens if you delay it?,” “Who is affected?”
  - Teacher collects 2-3 examples on the board to show how prioritization impacts outcomes and stakeholders, not just “boxes.”

- **Reflection (10–15 mins)** – Students apply prioritization to their own weekly goals.
  - Step 1: Students complete a blank matrix using personal or professional tasks.
  - Step 2: In pairs, share 1–2 surprising insights: “What did you realize about your priorities?”
  - Step 3: Whole group check-in: “What’s one thing you can delegate this week?”
  - **Materials:** Student blank matrix grid (Eisenhower Matrix)

[20-Minute Break]

III. Production (30–40 mins)

- **Role-Play: Team Planning Scenario (30–40 mins)** – Students simulate a team meeting where they must schedule and assign tasks.
  - Step 1: Provide a scenario: “Your team has 5 major tasks and a deadline in 3 days. Prioritize and delegate as a group.”

- Step 2: In groups of 3–4, students decide task order and assign responsibilities.
- Step 3: Each group presents its plan and explains reasoning.
- Teacher prompt: “How did you decide which task to do first?”
- **Materials:** Role-play Team Planning handout

IV. Digital Tool (45 mins)

- To provide students the opportunity to use the digital tool in class with teacher support.
- The teacher can demonstrate activities with the whole class and/or support students as they work individually.

V. Wrap-Up (15 mins)

- **Vocabulary Review (5–7 mins)**
  - Game: “This or That?” Teacher reads a sentence, Students respond orally by saying the correct word from two options you provide:
    - Example: “Delaying a task because it’s boring. Is this procrastinate or schedule?”
  - Teacher confirms the correct answer and provides a short explanation if needed.
  - Read the following prompts to continue the review:
    - “Organizing tasks in order of importance. Is this prioritize or ignore?”
    - “A goal that is Specific, Measurable, Achievable, Relevant, and Time-bound. Is this a SMART goal or random goal?”
    - “The quality of being responsible for your actions. Is this accountability or flexibility?”
    - “A significant stage or event in a project. Is this a milestone or detour?”
    - “Forward or onward movement toward a goal. Is this progress or delay?”
  - Teacher tips:
    - Encourage fast oral responses to keep the pace lively.
    - Ask students to use the word in a short sentence after confirming the answer.
    - Randomly call on different students to keep everyone engaged.

- **Self-Reflection (7–10 mins)**
  - Prompt students to write:
    - “What’s one change I can make to improve how I manage my time?”
    - “What’s a task I often treat as urgent but isn’t really important?”
  - Optional share with partner

Optional Independent Practice

- Try time-blocking one full day this week using Google Calendar or your preferred app.
- Bring a screenshot or summary to next class.
- Optional: Track how long common work tasks actually take.

Notes for the Instructor

- **Pacing Tip:** Keep controlled activities brisk to leave enough time for role-play and digital work.
- **Corrections Tip:** Focus corrections on functional language (e.g., justifications using “because,” “so that,” etc.).
- **Differentiation:** For students needing support, pre-fill part of the prioritization grid. For advanced learners, have them add unexpected workplace interruptions and re-prioritize.

- **Course Title:** English for Software Professionals **(Student Success Introductions)**
- **CEFR Level:** B2
- **Lesson Number:** 4
- **Topic:** Tools I Use and Why
- **Lesson Duration:** 3 hours (1hr20 - break 20mins - 1hr20)
- **Can-Do Objectives:** (Aligned with CEFR descriptors)
    - I can assess my progress towards language goals.
    - I can describe and evaluate learning tools I use.
    - I can justify my choice of tools and strategies using clear examples.

Materials

- Handouts:
    - Tool Evaluation Matrix Part
    - Tool Recommendation
    - Personal Development Plan

- Audio/Video Files:
    - Digital Learning Apps Video: I Tried 47 AI Language Learning Tools. These Are The Top 10 Best

- Required Tech / Supplies:
    - Whiteboard and markers
    - Projector

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
platform	A digital system or service designed for a specific function	I use this platform to manage my language learning and track progress.	Y
feature	A specific function or capability of a tool	One useful feature is real-time feedback on my writing.	Y
evaluate	To assess quality, effectiveness, or value using criteria	I evaluate learning tools based on usability and results.	Y
usability	How easy and efficient a tool is to use	The usability of this app makes it practical for daily work use.	Y

effectiveness	How well something produces the intended result	This tool is effective because it improves accuracy over time.	Y
criterion / criteria	A standard used to judge or evaluate something	One key criterion for me is how well the tool supports professional writing.	Y
recommend	To suggest something as suitable based on evaluation	I recommend this tool because it integrates well with my workflow.	Y
progress	Improvement measured over time	I can clearly see my progress by reviewing weekly reports.	Y

Lesson Structure (PPP)

- Warm-Up / Review: Review of Prioritization and Time Tools (10–15 mins)
    - Step 1: Ask: "What tools or systems did you identify as helpful in your learning last time?" Write student responses on the board.
    - Step 2: Distribute a short 5-question review worksheet based on Lesson 4 (see worksheet link). Example: “What tool helps you track your goals?”
    - Step 3: Pair-share: Students compare answers and explain choices.
    - Step 4: Teacher asks: "Have you tried using any new tools this week?" Encourage brief class discussion.

I. Presentation (30–40 mins)

- **Vocabulary Introduction (10 mins)** – Evaluating and Describing Tools
    - Step 1: Vocabulary focus (4-5 mins)
        - Write the eight vocabulary words on the board.
        - Encourage students to suggest quick workplace-based examples. Reformulate and extend their ideas to model accurate usage.
        - Use guiding questions instead of definitions to encourage deeper thinking:
            - Platform vs. feature – what’s the difference?
            - Usability – what makes one tool “easy to use”?
            - Evaluate – how would you evaluate a new business tool or software?
    - Step 2: Mini-task (5 mins)
        - In pairs, students choose one business tool they both use (for example: Slack, Asana, Microsoft Teams etc.). Each pair creates 2–3 sentences using at least three of the target vocabulary words.

        - Example: “The platform I use most is Slack. The usability is very high, and my favorite feature is creating private channels. I evaluate project updates every day.”
        - Pairs share their examples briefly with the class.
    - **Materials:** Whiteboard and markers

- **Grammar/Function Focus (10–15 mins)** – Describing purpose and justification (e.g., "I use X because..." / "This tool helps me... by...")
    - Step 1: Write and model a few sentence stems on board.
    - Step 2: Students complete 3 sentences in notebooks about their own tools.
        - Example: “I use Quizlet because it helps me remember new words by reviewing daily.”
    - Step 3: Pair up to practice describing tools out loud.
    - **Materials:** Whiteboard and markers

- **Mini-Lecture & Guided Discussion (10–15 mins)** – Explore categories of tools (e.g., flashcard apps, video platforms, group messaging) and how they support learning.
    - Step 1: Show video for 7-8min (I Tried 47 AI Language Learning Tools. These Are The Top 10 Best)
    - Step 2: Ask: “Which of these have you used? What’s your favorite feature?”
    - Step 3: Discuss: “Do you prefer digital or analog tools? Why?”

II. Practice (30–40 mins)

- **Controlled Activities (20 mins)** – Tool Evaluation Matrix
    - Step 1: Distribute a printed Tool Evaluation Matrix with columns: Tool, Purpose, Favorite, Feature, Rating (1–5), and Why?
    - Step 2: Students complete the grid for 2–3 tools they use.
    - Step 3:
        - Pair-share answers:
            - “Which tool would you recommend to a friend and why?”
    - **Materials:** Tool Evaluation Matrix

- **Reflection (10–15 mins)**
    - Step 1:
        - Write: “Which tool has helped your English most this year? How do you know?”
    - Step 2: Volunteers share reflections with the class.
    - Teacher prompt: “How do you evaluate your progress using this tool?”

[20-Minute Break]

III. Production (30–40 mins)

- **Tool Recommendation Pitch (30–40 mins)** – Students compare language learning tools, and present a short tool recommendation.
    - Step 1: Students complete a planning form (What is the tool? Who is it for? Why is it effective?).
        - Additional requirement: students must include at least one evaluation criterion (effectiveness, usability, feature, etc.)
    - Step 2: Teacher model
        - Model a recommendation using one tool. Example:
            - "Today I'm recommending Grammarly, a language platform for professionals who write frequently. It's effective because it gives real-time feedback and tracks accuracy over time. I've evaluated it based on usability and adaptability, and it integrates easily with email and Word documents. For these reasons, I think it's a valuable tool for workplace communication."
    - Step 3: Present to a partner or small group.
    - Step 4: Listeners write 1 follow-up question for the speaker.
    - Teacher prompt: "What makes a tool worth recommending?"
    - **Materials:** Tool Recommendation Handout

IV. Mini-Project: Personal Development Plan (30–40 mins)

- **Personal Development Plan Project** – Students synthesize all learning from Unit 1 to create a detailed, personalized learning plan.
    - Step 1: Distribute planning worksheet with four sections: Learning Strategies, SMART Goals, Time Management Tools, and Evaluation Methods.
    - Step 2: Students complete the plan based on what they've learned in Lessons 1–4.
    - Step 3: Pair-share for feedback: "Does your plan match your learning style and goals?"
    - Step 4: Volunteers present highlights of their plan to the class.
    - Teacher prompt: "What do you think will be the hardest part of sticking to your plan? How will you stay on track?"
    - **Materials:** Personal Development Plan

V. Wrap-Up (15 mins)

- **Vocabulary Review (5–7 mins)**
    - Matching game on board: terms and definitions
    - Optional: write a sentence using 2 words
    - **Materials:** Whiteboard and markers

- **Self-Reflection (7–10 mins)**
    - Write: "What tool will I start (or continue) using and how will I track my results?"

- Share one response with a partner.

Optional Independent Practice

- Choose a tool you haven't used before. Try it 2–3 times this week.
- Complete a short reflection or record an audio message on your experience.

Notes for the Instructor

- **Pacing Tip:** Monitor time during pitch prep to ensure students have enough time to share.
- **Corrections Tip:** Encourage clarity when describing tools (watch out for overuse of vague terms like "thing" or "stuff").
- **Differentiation:** Offer lower-level students a sentence frame bank for pitches; encourage advanced learners to compare multiple tools.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: 1
Topic: Describing Your Role and Experience in Depth
Lesson Duration: 3 hours (1hr20 – break 20mins – 1hr20)
Can-Do Objectives (Aligned with CEFR descriptors):

- Can describe their role in detail, including key tasks and tools
- Can highlight results or contributions using "I was responsible for…"
- Can present their work history in a structured way

Materials

- Icebreaker "Find Someone Who…" sheet
- Guided worksheet: "My Role – Responsibilities & Impact"
- Sample LinkedIn profiles (anonymized open-access examples provided as text)
- Sentence frame sheet ("I was responsible for…", "I implemented… which led to…")
- Whiteboard and markers
- Projector or screen

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
responsibilities	tasks that someone is required to do	"My responsibilities include testing and reviewing code."	N
contribute	to give or add value to a project	"I contributed to developing the payment system."	N
impact	the effect or result of work	"My work had a big impact on reducing bugs."	N
collaborate	to work together with others	"I collaborate with designers and testers daily."	N
implement	to put a plan or design into action	"I implemented a new login feature."	N
maintain	to keep something working properly	"I maintain the company's internal database."	N
streamline	to make a process more efficient	"We streamlined the workflow to save time."	N
oversee	to supervise or manage something	"I oversee a team of junior developers."	N

deploy	to release software to users	“We deploy updates every two weeks.”	N
outcome	the result of a project or task	“The outcome of our sprint was a new feature release.”	N

Lesson Structure (PPP)

Warm-Up Icebreaker (20 mins)

- **Course Overview (5 mins)**
  - Teacher introduces course goals: “By the end, you will be able to present systems, lead meetings, write professional documentation, and have in-depth discussions in English.”
- **Icebreaker “Find Someone Who…” (15 mins)**
  - Handout: Grid with prompts (e.g., “Has worked with Python,” “Has deployed code to production,” “Has worked in a team of 10+”).
  - Students walk around, asking classmates questions to find matches. Must speak only in English.
  - Scaffold: Provide question frames on board: “Have you ever…?”, “Do you usually…?”, “What tools do you use?”
  - Debrief: Students share one interesting fact they learned about a classmate.

I. Presentation (30 mins)

Vocabulary Introduction (15 mins)

- Teacher introduces 10 words from the table.
- Worksheet: Students work in pairs to match vocabulary to definitions, fill in the blanks, etc.

Function/Grammar Focus (15 mins)

- Teacher models how to structure role descriptions, write on the board:
  - “I was responsible for ___.”
  - “I implemented ___, which resulted in ___.”
- Show 2 short sample LinkedIn profile snippets (included in handout).
- Model how to underline action verbs and outcomes
- Students work in pairs to underline action verbs + outcomes by themselves

II. Practice (40 mins)

Controlled Practice: My Role Worksheet (20 mins)

- Handout prompts students to list:
  - 3–4 responsibilities
  - Tools or languages used
  - 1 outcome/impact for each
- Scaffold: Provide sentence frames:

    - I was responsible for ___, which led to ___.
    - I implemented ___ using ___.
    - I collaborated with ___ to ___.
    - The outcome was ___.

- Pair-check responses, teacher highlights strong examples.

Pair Work: Mini Interview (20 mins)

- Role cards:
    - Interviewer: Ask “What are your main responsibilities?” “What tools do you use?” “What’s an achievement you’re proud of?”
    - Interviewee: Answer using vocabulary + sentence frames.
- Switch roles. Teacher circulates, noting errors and good phrases.

[20-Minute Break]

III. Production (40 mins)

Drafting a Short Professional Profile (20 mins)

- Tell students they will write a 3–4 sentence LinkedIn-style description of their role (not full profile yet).
    - Include: job title, 2 responsibilities, 1 outcome/achievement.
    - Use at least 5 vocabulary words.
- Before the students start, go onto linkedin on the projector and look at a few people’s role descriptions. Point out to students how the verbs are used in present tense

Peer Feedback (20 mins)

- Students exchange drafts, underline action verbs, and check for outcomes.
- Checklist provided:
    - Are responsibilities clear?
    - Is at least 1 outcome included?
    - Are action verbs used?
- Students refine drafts. Teacher selects a few to read aloud (with permission).

IV. Wrap-Up (15 mins)

Vocabulary Review Game (5 mins): “Action Verb Relay”

    - Put students into small groups (3-4)
    - Teacher calls a word
    - Teams race to produce a correct sentence.
    - Keep score to create some competition, or not if based on the class

Self-Reflection (5 mins):

    - Write on the board: “Today I learned how to describe…”

    - Students turn to a partner and complete the sentence
    - Ask a few students to tell the class their answers

Optional Independent Practice

- Explore LinkedIn (or similar open-access job board profiles). Find one role similar to yours. Note:
    - 3 action verbs you could use
    - 1 way they describe their impact
- Optional (if access available): Update your own LinkedIn profile with today's vocabulary.

Notes for Instructor

- Be mindful this is the first content class → keep energy high and focus on student interaction.
- Scaffold for less confident students with sentence frames and model answers.
- Highlight professional, confident language ("I implemented X which improved Y") over vague language ("I did some coding").
- Encourage students to notice style differences between their writing and authentic LinkedIn profiles.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: 2
Topic: Identifying and Articulating Strengths
Lesson Duration: 3 hours (1hr20 – break 20mins – 1hr20)
Can-Do Objectives (Aligned with CEFR descriptors):

- Can identify their professional strengths and explain how they use them
- Can say: "One of my strengths is…" and support it with an example
- Can talk about challenges they've overcome

Materials

- "Strengths Brainstorm" worksheet
- "STAR Method" guide + worksheet (Situation, Task, Action, Result)
- Role-play cards: common interview-style strength questions
- Peer feedback checklist for mini-presentations
- Whiteboard or shared doc for collecting student strengths
- Projector/screen for showing sample video

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
strength	a quality or skill you are good at	"One of my strengths is solving complex problems."	N
weakness	an area that needs improvement	"My weakness is time management, but I'm improving."	N
achievement	something you have done successfully	"A major achievement was launching our app on time."	N
persistence	continuing even when things are difficult	"My persistence helped me fix a critical bug."	N
adaptability	ability to change when new situations arise	"Adaptability is key when new frameworks appear."	N
leadership	ability to guide and inspire others	"My leadership skills helped the team stay motivated."	N
problem-solving	ability to find solutions to challenges	"Problem-solving is essential in debugging."	N
initiative	ability to act without being told	"I took initiative to improve our testing process."	N

communication	ability to share ideas clearly	“Good communication avoids misunderstandings.”	N
resilience	ability to recover quickly from difficulties	“Resilience is important after project setbacks.”	N

Lesson Structure (PPP)

Review (10 mins): “Compliment Chain”

- Ask students to get into groups of 4-5 in a circle
- Step 1: In a circle, Student A turns to Student B and says one professional compliment using last lesson’s vocabulary (e.g., “You collaborate well with your team”).
- Step 2: Student B thanks them and then gives a compliment to Student C, and so on.
- Scaffold: Provide compliment starters on board:
  - “I think you are good at…”
  - “I noticed in our last activity you ___.”
- Purpose: Recap action verbs and start thinking positively about skills/strengths.

Warm-Up (10 mins)

- Ask students to turn to a partner and discuss a professional challenge they’ve experienced this year.
- One students speaks about a challenge, the other asks clarifying questions, then switch

I. Presentation (30 mins)

Vocabulary Introduction *(15 mins)*

- Teacher introduces today’s 10 words with definitions + examples.
- **Worksheet** matching: term → definition → example sentence.
- Students personalize by writing 1 new sentence for 3 different words about themselves.
- Teacher walks around and makes sure students are on track, go over as a class if necessary.

“Strengths in Action” (15 mins)

1. Write these prompts on the board or slide:
   - What is one of your professional strengths?
   - When have you used it recently?
   - What was the result or impact?
2. **Model with a simple example:**
   - “One of my strengths is *problem-solving*.
     Last week our app had a bug in the login process.
     I found the error in the API request and fixed it, and the system ran smoothly after.”

3. **Pair Practice: Put students in pairs**
    - Students think of one of their own strengths.
    - Partner A asks: “What’s one of your strengths?”
    - Partner B answers using the prompts above
    - Switch roles after 2 minutes.
4. **Optional Challenge:**
   Have a few students share their examples aloud. Encourage use of connectors like *because*, *so*, *as a result*, *which helped…*

II. Practice (40 mins)

Strengths in Real Situations (20 mins)

1. Distribute the worksheet or display this prompt:
    - Think of 3 professional strengths you use at work.
    - Choose one and describe a short situation where you used it successfully.
2. Scaffold with a simple table (students fill it in):

My Strength	Example Situation	What I Did	What Happened (Result)
Problem-solving	Our login feature was crashing	I checked the error logs and fixed the API bug	The system became stable again

3. **Sentence Frames to Support:**
    - “One of my strengths is ___.”
    - “I used it when ___.”
    - “I did ___ to solve the issue.”
    - “As a result, ___.”

4. **Pair Sharing (5–7 minutes):**
   Students take turns sharing one strength and example with a partner.
   Partners ask one follow-up question:

   “How did that help your team?” or “What did you learn from that?”

5. **Optional Extension (3–5 minutes):**
   Volunteers share one example aloud.
   The teacher highlights strong connectors (*as a result, because, so, which helped*).

Role-Play Cards: Interview Practice (20 mins)

- Pass out Role Play Cards which include common prompts:
    - “Tell me about one of your strengths.”
    - “Give an example of a challenge you overcame.”
    - “Describe a time you showed leadership.”
- Students work in pairs: Student A = interviewer, Student B = interviewee.

- Then switch roles.
- Teacher monitors and notes strong examples/errors.
- At the end ask students to present a short exchange for the class

[20-Minute Break]

III. Production (40 mins)

Mini Presentation: "My Top Strength" (20 mins)

- Students prepare a 2-minute talk:
  - Introduce 1 strength
  - Give 1 thing they want to prove it
- Students present to partner or small group.

Peer Feedback + Refinement (20 mins)

- Use checklist:
  - Did they clearly state the strength?
  - Did they give a real example?
  - Did they explain the result/outcome?
- Students give one compliment + one suggestion.

IV. Wrap-Up (15 mins)

Vocabulary Review Game (5 mins): "Strength Snap"

  - Teacher says a strength; students "snap" fingers if it applies to them. Volunteers explain why.
- Self-Reflection (5 mins):
  - Students write: "One strength I want to use more is…"

Optional Independent Practice

- Explore Glassdoor or LinkedIn "common interview questions."
- Write down 2 strengths you might use in an interview
- Optional: If you have LinkedIn access, update your profile "About" section with one strength + example.

Notes for Instructor

- Encourage specificity: "I am good at coding" → too general. "I am good at debugging complex back-end issues" → strong.
- For weaker students, allow them to start with just "strength + sentence."
- Model vulnerability too: acknowledging a challenge/weakness is realistic and helps students sound more authentic.
- Praise risk-taking when students share personal or professional examples.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: 3
Topic: Comparing Roles, Tools, and Specializations
Lesson Duration: 3 hours (1hr20 – break 20mins – 1hr20)
Can-Do Objectives (Aligned with CEFR descriptors):

- Can compare two tools or roles using advanced comparisons
- Can express preferences and tradeoffs: “Although X is faster, Y is more scalable”
- Can use linking language like “whereas,” “while,” and “on the other hand”

Materials

- Vocabulary matching worksheet
- Comparison sentence frame handout (“whereas,” “on the other hand…”)
- Case study cards (Frontend vs Backend, Java vs Python, Agile vs Waterfall, etc.)
- Peer feedback checklist for debates/discussions
- Required Tech:
    - Whiteboard/markers or digital board for comparing pros/cons
    - Projector/screen to show video or comparison tables

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
frontend	part of software users see and interact with	“Frontend developers focus on UI/UX.”	Y
backend	server-side logic, database, APIs	“Backend developers manage data flow.”	Y
scalability	ability to handle growth	“Kubernetes improves scalability.”	Y
performance	how fast/efficient a system runs	“The new code increased performance by 20%.”	Y
usability	how easy something is to use	“Frontend changes improved usability.”	Y
maintainability	ease of updating/fixing code	“Readable code increases maintainability.”	Y
trade-off	balancing one benefit against another	“There’s a trade-off between speed and security.”	Y
versatility	ability to be used in many ways	“Python’s versatility makes it popular.”	Y

complexity	degree of difficulty	“The system’s complexity slowed onboarding.”	Y
reliability	consistency of performance	“Reliability is crucial for financial software.”	Y

Lesson Structure (PPP)

Review (10 minutes)

- Ask students to turn to a partner and go over the strengths they wrote down from last class, encourage them to share their linkedIn with their partners to get feedback and go over previous language from last lesson.

Warm up(15 mins): “Pros and Cons Brainstorm”

- Teacher writes “Frontend vs Backend” on board.
- Small groups brainstorm 3 pros and 3 cons for each. (if time doesn’t allow, make 1-2 pros and cons)
- Groups share → teacher highlights linking language: “whereas,” “on the other hand,” “in comparison.”

Frontend — Pros

1. *More visible impact — You can immediately see the results of your work in the UI.*
2. *Creative problem-solving — It involves design, user experience, and interaction.*
3. *Direct user feedback — Users notice changes quickly and give fast reactions.*

Frontend — Cons

1. *Frequent changes — UI trends and frameworks change constantly.*
2. *Browser inconsistencies — Features behave differently across browsers.*
3. *High pressure from users — Mistakes are noticeable immediately.*

Backend — Pros

1. *Deep technical logic — You work on system architecture, data processing, APIs.*
2. *More stability — Backend frameworks change less often than frontend ones.*
3. *Higher system impact — Good backend design improves performance and scalability.*

Backend — Cons

1. *Less visible work — Your contributions are not always seen by end users.*
2. *Complex debugging — Problems may come from multiple services or dependencies.*
3. *Harder to explain to non-technical teams — The logic is abstract and not visual.*

I. Presentation (30 mins)

Vocabulary Introduction (15 mins)

- Students work in pairs
- Worksheet: Students match today’s terms with definitions and example sentences.
- Quick check: Students create 1 new sentence with a term, connected to their own work.

Grammar/Function Focus (15 mins)

- Teacher models advanced comparison sentences. Write on the board:
    - “Although Java is more reliable, Python is more versatile.”
    - “Frontend focuses on usability, whereas backend focuses on scalability.”
- Scaffold: Provide comparison sentence frames:
    - “While ___ is ___, ___ is ___.”
    - “Compared to ___, ___ is ___.”
    - “On the other hand, ___.”
- Ask students to work in pairs and practice creating advanced comparison sentences together using the sentence frames.
- Encourage students to write down 2-3 strong sentences and to practice until they can say them without reading off the paper.

II. Practice (40 mins)

Controlled Practice: Sentence Expansion (20 mins)

- Handout: pairs of items (e.g., Python vs Java, Frontend vs Backend, Agile vs Waterfall).
- Students write 3 full comparison sentences for each pair using linking words.
- Share in pairs, then go around the room as a class and ask pairs to share

Case Study Cards: Role/Tool Debate (20 mins)

- Students work in groups of 4.
- Each group gets a card (examples):
    - “Your team must choose: Python or Java for a new app.”
    - “You are hiring: frontend vs backend developer.”
- Groups split into two teams and choose which side they argue for
- Teams prepare arguments, then hold a debate.
- Teacher circulates and notes strong linking language.

[20-Minute Break]

III. Production (40 mins)

Mini Debate: “Which is Better?” (30 mins)

- Teacher gives a statement (choose one based on your students or give them options to choose from
    - Agile is always better than Waterfall.
    - “Pair programming is better than working independently.”
    - “Remote teams are more productive than in-office teams.”
    - “Code reviews should be strict rather than flexible.”
    - “Daily stand-ups should be required for all teams.”
- Class splits into 2 sides, prepare 3 arguments each, and debate.
- Scaffold: Give debate sentence starters on board:
    - “On the other hand…”
    - “Compared to…”

    - “Although…”

Peer Feedback + Reflection (10 mins)

- Students use checklist to evaluate peers:
    - Did they use linking words?
    - Did they state preferences clearly?
    - Did they explain a trade-off?

IV. Wrap-Up (15 mins)

Vocabulary Review Game (5 mins): “Hot Seat”

- One student sits facing class. Teacher writes words on the board for the class to see
- Class must explain the word ot the student without using the word itself
- Self-Reflection (5 mins):
    - Students write: “One comparison I can now make is…”

Optional Independent Practice

- Visit Stack Overflow or a developer blog. Find a discussion comparing two tools or frameworks. Copy 1 comparison sentence and rewrite it in your own words.
- Optional: Post a comment or note comparing two tools you use in your work.

Notes for Instructor

- Encourage precision — avoid generalities like “Python is good.” Push for “Python is versatile, whereas Java is reliable.”
- Support less confident students with sentence frames before debates.
- For advanced students, push longer comparisons (2–3 trade-offs in one response).
- Keep debates structured and time-limited to maximize speaking opportunities.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: 4
Topic: Creating a Portfolio Pitch
Lesson Duration: 3 hours (1hr20 – break 20mins – 1hr20)
Can-Do Objectives (Aligned with CEFR descriptors):

- Can structure a 3–5 minute presentation with a clear flow
- Can emphasize key experiences and tailor them to an audience
- Can answer follow-up questions with confidence

Materials

- Portfolio Pitch Planning Worksheet (outline + sentence frames)
- Example Portfolio Pitch transcript (authentic-style model)
- Phrases sheet: "Highlighting experience and results" (anchored to reading)
- Audience Question Cards
- Peer Feedback Checklist for final presentations

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
portfolio	a collection of work samples or projects	"I showed my portfolio to the hiring manager."	Y
highlight	to emphasize something important	"I highlighted my leadership on the mobile app team."	N
relevance	importance to the situation	"Choose examples with the most relevance to the role."	N
tailor	to adapt something for a purpose or audience	"I tailored my pitch for non-technical recruiters."	Y
achievement*	something completed successfully	"A major achievement was deploying the app early."	Y
flow	smooth order/connection of ideas	"Her presentation had a logical flow from start to end."	N
concise	short but clear	"Keep your portfolio pitch concise and focused."	N
confidence	the quality of speaking strongly and clearly	"He spoke with confidence about his projects."	N

engage	to capture interest	“He engaged the audience by asking a question.”	N
Q&A (questions and answers)	interactive question section after a presentation	“The Q&A gave me a chance to clarify details.”	Y

Lesson Structure (PPP)

Review

“Compare and Decide” – Pair Practice

1. Teacher writes on the board:
   Frontend vs Backend
   SQL vs NoSQL
   Standalone app vs Microservice
2. Students work in pairs and choose one pair to compare.
3. They must produce two comparative sentences using linking language:

Write on the board:

- “Whereas…”
- “On the other hand…”
- “In comparison…”
- “Although…”

Example Model Answers (for teacher):

- “Frontend developers focus on user interfaces, whereas backend developers work with logic and data flow.”
- “SQL databases are structured, on the other hand NoSQL databases are more flexible.”

4. Pairs share one example with the class.

Warm-Up (10 mins)

3 Facts About My Experience – Pair Exchange

- Students think of three facts about their professional experience:
  - one responsibility
  - one tool they use
  - one project or task they contributed to

2. Student A shares their three facts orally.
3. Student B asks a follow-up question for each fact.

Follow-Up Question Starters (on board):

- “Can you explain how you used…?”
- “Why was that important for the project?”
- “What was the most challenging part?”

After 2 minutes, students switch roles.

I. Presentation (30 mins)

Vocabulary Anchored in Input (15 mins)

- Read the **Model Portfolio Pitch Transcript**
- Students underline vocabulary words (highlight, tailor, achievement, relevance, flow, engage).
- Comprehension Questions:
    1. How did the speaker structure their talk?
    2. Which projects did they highlight?
    3. How did they adapt to their audience?
- Teacher reviews answers, linking each to key vocabulary in context.

Phrases in Context (15 mins)

- Teacher projects sentences from the model text:
    - "One achievement I'd like to highlight is…"
    - "I tailored this project for non-technical clients."
    - "I'm proud that it improved scalability by 40%."
- Students write down and then personalize in pairs:
    - Partner A: "One project I'd like to highlight is ___."
    - Partner B: "What problem did it solve?" / "Who was the audience?"
- Variation: switch and add one more linking word (because, which led to, as a result).

II. Practice (40 mins)

Controlled Practice: Portfolio Pitch Planner (20 mins)

- Handout: **Portfolio Pitch Planning Worksheet**
- Students outline a 3–5 minute pitch using these prompts:
    1. Who am I? (role, experience, specialization)
    2. What 2–3 projects will I highlight?
    3. What skills or results do they show?
    4. What's relevant for my audience (recruiter, manager, team)?
    5. What's my closing message?
- Scaffold: sentence frames included (e.g., "A key achievement in my current role is…", "This project is especially relevant because…").

Pair Work: Rehearsal & Timing (20 mins)

- Students deliver draft pitch to partner (2–3 minutes).
- Partner uses checklist to give immediate feedback on:
    - Clarity of structure
    - Use of linking language
    - Balance of detail and conciseness
- Switch roles.

[20-Minute Break]

III. Production (40 mins)

Mini-Project: Portfolio Pitch Presentation (30 mins)

- Students deliver a 3–5 minute portfolio pitch to a small group (3–4 students).
- Must include:
    - Introduction to current role and experience
    - A detailed project example
    - Reflection on strengths and goals
    - A short Q&A session with peers or instructor
- Each group selects one presenter for a final in-class presentation.

Peer Feedback + Reflection (10 mins)

- Peers complete **Feedback Checklist**:
  ✓ Clear structure and flow
  ✓ Relevant examples highlighted
  ✓ Confident delivery
  ✓ Questions answered clearly
- Students write one compliment and one improvement goal.

IV. Wrap-Up (15 mins)

Vocabulary Review Game (5 mins): "Pitch Bingo"

- Teacher lists 8 words (portfolio, highlight, tailor, engage, relevance, achievement, confidence, concise).
- Students listen to peers' mini-presentations and mark words they hear.

Optional Independent Practice

- Watch one short portfolio or career presentation on YouTube.
    - Identify how the speaker structured their talk (intro → examples → closing).
    - Note 2 phrases you could use in your own presentation.
- Optional: Record a 2–3 minute version of your pitch and self-evaluate using the peer checklist.

Notes for Instructor

- Encourage students to speak from notes, not read, aim for confidence and eye contact.
- Keep focus on clarity and audience awareness rather than accent or speed.
- Time presentations carefully; give verbal cues for remaining time.
- Offer gentle correction only if communication is blocked; otherwise, focus feedback on fluency and relevance.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: 5
Topic: Expressing Opinions and Reaching Consensus
Lesson Duration: 3 hours (1hr20 – break 20mins – 1hr20)
Can-Do Objectives (Aligned with CEFR descriptors):

- Can express opinions confidently and give reasons
- Can agree or disagree politely and constructively
- Can help a group reach a decision by summarizing options

Materials

- Google Docs – Shared "Team Decision Log" template for group consensus tasks
- Whats app – For digital chat simulation (agree/disagree discussion thread)
- Mentimeter – For warm-up and decision-making polls
- Miro – For visual decision maps during practice
- Shared Phrases Sheet (digital comments enabled for annotation)
- Peer Feedback Form (Google Form or editable doc)

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
consensus	general agreement among a group	"The team reached consensus after a short discussion."	N
compromise	an agreement where each side gives up something	"We found a compromise that satisfied both developers."	N
justify	to explain the reason for a decision	"Can you justify why you prefer React over Vue?"	Y
perspective	a particular point of view	"From a design perspective, it's easier to use Figma."	N
feedback	helpful information about someone's work	"We gave feedback on each other's pull requests."	Y
constructive	helpful and positive in tone	"She gave constructive criticism on the code style."	Y
diplomatic	polite and careful in expressing disagreement	"He disagreed diplomatically with his manager."	N
evaluate	to assess or judge carefully	"Let's evaluate both options before deciding."	Y

alternative	another possible choice or solution	“We discussed several alternatives to improve performance.”	Y
summarize	to restate the main points briefly	“Let me summarize what we’ve agreed on so far.”	N

Lesson Structure (PPP)

Review (10 mins): “Portfolio Highlights Recap”

- Students open their digital portfolio pitch notes from Lesson 4.
- In breakout pairs, they share one project they presented last time and add one short sentence giving their opinion about it:
  - “I think this project was my best work because…”
  - “I prefer this project since it showed my teamwork.”
- Teacher highlights natural transitions from describing to expressing opinions — bridging the previous unit to the new one.

Warm-Up (10 mins): “Opinion Poll – Tech Preferences” (Digital Poll)

- Students join a **Mentimeter** with 3 opinion-based questions:
  - “Which framework do you prefer: React, Vue, or Angular?”
  - “Which communication tool is best for teamwork?”
  - “Which language is most efficient?”
- Display live results and briefly discuss:
  - “Do you agree with the majority?”
  - “Why or why not?”
- Transition: “In today’s lesson, we’ll focus on how to express these kinds of opinions clearly and diplomatically in a team discussion.”

I. Presentation (30 mins)

Vocabulary Intro (10 min)

- Send the vocabulary table to the class group chat
- Have students work alone or in pairs to review the words, write down any words they don’t know yet and write 3-4 sentences using the vocabulary words
- Ask students to share their example sentences with the class

Listening Input: Tech Team Debate (10 mins)

- Play short video clip of developers discussing tool choices (or use a written transcript).
- Students in breakout pairs note 3 expressions used to give opinions or agree/disagree.
- In a shared doc, students highlight examples (anchored language).
- Debrief: teacher elicits and organizes phrases under 3 headings on shared board:
  - Giving Opinions
  - Agreeing / Disagreeing

    - Reaching Consensus

Phrases in Context (10 mins)

- Teacher shares **Digital Phrases Sheet**:
    - Giving opinions: “I think…,” “From my perspective…,” “In my experience…”
    - Agreeing: “I completely agree,” “That’s a good point,” “Exactly.”
    - Disagreeing diplomatically: “I see your point, but…,” “I’m not sure I agree with that because…”
    - Summarizing / Closing: “So, we all agree that…,” “Let’s move forward with…”
- Students annotate phrases with comments → examples from their own work.
- Pair Practice (in chat simulation): Students role-play a Slack thread debating which framework to use.

II. Practice (40 mins)

Jigsaw Discussion (20 mins)

- Divide class into 3 “expert groups”:
    - Group A: prefers React
    - Group B: prefers Vue
    - Group C: prefers Angular
- In breakout rooms, groups list 3 pros and cons using a **shared Miro board**.
- Re-group into mixed teams → share and debate which is best for a new app project.
- Each group uses the “Team Decision Log” Google Doc to record arguments and final consensus.

Consensus Challenge (20 mins)

- Scenario (digital doc provided):
    - “Your company needs to choose a collaboration tool. The options: Slack, Microsoft Teams, or Discord.”
- Groups discuss via breakout or shared doc comments.
- Goal: reach consensus and record final statement beginning with “We all agree that…”

[20-Minute Break]

III. Digital Tool (45 mins)

- To provide students the opportunity to use the digital tool in class with teacher support.
- The teacher can demonstrate activities with the whole class and/or support students as they work individually.

IV. Wrap-Up (15 mins)

Reflection (10 mins): “From Disagreement to Decision”

- Students answer 3 reflection prompts in chat or shared doc:

    - “Which phrase helped you sound more diplomatic?”
    - “How did your group reach consensus?”
    - “What was hardest about expressing your opinion in English?”

Mentimeter Exit Poll (5 mins)

- “How confident are you now at expressing opinions in meetings?” (1–5 scale).
- Display results → brief verbal wrap-up.

Optional Independent Practice

- Join an open-source forum (e.g., Stack Overflow or GitHub Discussions).
- Read one short debate thread about a tech choice.
- Identify 2 polite ways people agree/disagree.
- Write a short reflection (3 sentences) in English about which approach sounds most diplomatic.

Notes for Instructor

- Encourage turn-taking and balanced participation, engineers may default to “listener mode.”
- Highlight polite tone markers (“I see your point, but…”) as essential for international teams.
- Use real-world tools (Slack or Discord) whenever possible to simulate genuine communication flow.
- Emphasize clarity + diplomacy, not just correctness.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: 6
Topic: Giving and Receiving Feedback
Lesson Duration: 3 hours (1hr20 – break 20mins – 1hr20)
Can-Do Objectives (Aligned with CEFR descriptors):

- Can give specific, balanced feedback using frameworks like "What worked / What to improve."
- Can respond to feedback without becoming defensive.
- Can revise a message or task based on feedback.

Materials

- **Google Docs / Notion** – shared document for peer feedback comments.
- **Slack or Discord** – channel simulation for giving written feedback.
- **Miro** – "Feedback Framework" board (What Worked / What to Improve).

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
feedback*	helpful information about work or performance	"We gave feedback after each sprint review."	Y
constructive*	useful and positive in tone	"Her comment was constructive, not critical."	N
defensive	reacting negatively to criticism	"Try not to sound defensive when you get feedback."	N
framework	structured way of doing something	"We use a feedback framework: 'What worked / What to improve.'"	Y
actionable	possible to act on or implement	"Give feedback that's specific and actionable."	Y
tone	the style or feeling of communication	"Written feedback should keep a neutral tone."	N
clarity	the quality of being clear and easy to understand	"I appreciated the clarity of her feedback."	N
empathy	understanding another person's feelings	"Show empathy before giving suggestions."	N
revision	a change made to improve something	"I made revisions after my manager's comments."	Y

openness	willingness to accept feedback	“Openness to feedback is important in agile teams.”	N

Lesson Structure (PPP)

Review (10 mins): “Consensus Check-In”

- Students open the shared Padlet from Lesson 5 (final decisions).
- In pairs, they give one positive comment and one suggestion on another group’s decision statement.
  - “I liked how clear your summary was.”
  - “Maybe you could explain why React was chosen.”
- Teacher highlights polite, professional phrasing.

Warm-Up (10 mins): “Feedback Attitudes Poll”

- Write on the board:
  - “How comfortable are you giving feedback?” *Define “Feedback” if necessary
  - Have students give thumbs up, thumbs down, or sideways to show their answers
  - Ask students “What kind of feedback helps you improve most?” Look for answers like “constructive feedback, positive feedback, polite feedback, specific feedback
  - Ask students “Why is feedback important in tech teams?” Discuss as a class.

I. Presentation (30 mins)

Vocabulary in Context (15 mins)

- Project the following dialogue on the board, have students work in pairs to read it together and then outloud practicing.

  Sam: The new API function works well, but the comments need more clarity.
  Dana: Thanks — I’ll revise those. Anything else?
  Sam: One suggestion — the naming could be more consistent.
  Dana: Good point. I’ll fix that before merging.
- Teacher elicits meanings for *clarity, revise, suggestion, feedback.*
- Quick interactive quiz (Mentimeter “Word in Context”) to confirm understanding.

Phrases for Giving & Receiving Feedback (15 mins)

- Phrases appear directly from the transcript; teacher groups them:
  - **Giving feedback:** “One suggestion is…,” “You might want to…,” “It could be clearer if…”
  - **Receiving feedback:** “Thanks for pointing that out,” “That’s a good idea,” “I’ll revise that.”
- In pairs, students practice short exchanges using shared Slack simulation thread:
  - A: “The diagram is clear, but maybe add a legend.”
  - B: “Good idea, I’ll add that before submission.”

II. Practice (40 mins)

"Feedback Framework Board" (20 mins)

- Teacher opens a **Miro board** with two columns:
  - *What Worked* | *What to Improve*
- Each pair receives a short sample text (e.g., a bug report or short feature description).
- Students write at least one sticky note for each column.
- Whole class debrief: discuss which comments are most actionable.

"Slack Feedback Simulation" (20 mins)

- Have a student open a class Slack/Discord channel "#code-review-practice."
- Ask a student to post a sample screenshot / code snippet / mock feature.
- Students post feedback messages using today's phrases.
- Partners reply using receiving-feedback language ("Thanks! I'll update…").
- Teacher highlights tone and diplomacy in real time.

[20-Minute Break]

III. Production (40 mins)

Mini-Project Simulation: Peer Code-Review Feedback (30 mins)

- In pairs students exchange a short written piece from previous lessons (portfolio pitch, project description, or meeting summary).
- Using the **Google Docs commenting feature**, each student gives:
  - Two *What Worked* comments.
  - One *What to Improve* comment.
- Then they read feedback received and write a 1-sentence reflection:
  "The most helpful feedback I got was ___ because ___."
- Teacher checks for tone, specificity, and professionalism.

Peer Debrief (10 mins)

- Volunteers share examples of constructive vs vague feedback.
- Class votes via Whatsapp poll: "Which comment was most actionable?"

IV. Wrap-Up (15 mins)

Reflection:

- In a shared document, students complete:
  - "Today I learned to…"
  - "Next time, I'll try to…"

Quick Game (5 mins): "Tone or Not?"

- Teacher reads feedback lines aloud (mix of polite/impolite).
  - "You should fix this — it's wrong." → ❌

    - “You might consider revising this part.” → ✔️
    - “You might consider adding a clearer example here.” ✔️
    - “This is wrong — fix it.” ❌
    - “Could we reorganize this part for better flow?” ✔️
    - “I don’t understand what you wrote. Rewrite it.” ❌
    - “Maybe we can simplify this explanation to help the reader.” ✔️
    - “Why would you do it like that?” ❌
    - “One suggestion is to break this into two steps.” ✔️
    - “This part is confusing. Redo it.” ❌
- Students vote via thumbs up or thumbs down.

Optional Independent Practice

- Choose one real piece of online feedback (e.g., GitHub pull-request comment).
- Copy a short example of constructive wording.
- Write 2–3 sentences explaining why the tone is effective.
- Post reflection in shared class doc under *Lesson 6 – Independent Practice.*

Notes for Instructor

- Reinforce the difference between *constructive* vs *critical* tone.
- Monitor written feedback for politeness and specificity.
- Highlight useful language that appears naturally in students’ posts.
- Encourage engineers to think of feedback as a technical skill, not a personal critique.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 7
Topic: Communication Channels and Tone
Lesson Duration: 3 hours (1hr20 – break 20mins – 1hr20)

Can-Do Objectives (Aligned with CEFR descriptors):

- Can choose the right tool for different types of messages
- Can adjust tone depending on the recipient and context
- Can revise a message for clarity, politeness, or directness

Materials

- Slack / Discord – written tone simulation for chat vs. formal email.
- Google Docs / Notion – "Tone Adjustment Board" (shared message-editing doc).
- Mentimeter / Slido – warm-up poll and wrap-up reflection.
- Miro / Jamboard – tone ranking board (formal ↔ informal scale).

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
tone*	the feeling or attitude in a message	"Her email had a friendly tone."	N
context	the situation or setting of communication	"In a formal context, avoid slang."	N
clarity*	being easy to understand	"He revised his message for clarity."	N
recipient	the person receiving a message	"Consider your recipient before you write."	N
directness	expressing ideas clearly and efficiently	"Developers value directness in bug reports."	Y
politeness	showing respect and courtesy in language	"Politeness is important in emails to clients."	N
medium	the channel or format of communication	"Choose the right medium: chat, email, or call."	Y
miscommunication	failure to understand correctly	"Tone differences can cause miscommunication."	N
nuance	subtle meaning or feeling	"Emoji can add nuance to written messages."	N

professionalism	appropriate behavior or tone at work	“Professionalism means balancing clarity and politeness.”	N

Lesson Structure (PPP)

Review (10 mins): “Feedback Reflection”
Students open their peer feedback comments from Lesson 6.
They choose one piece of feedback they received and discuss in pairs:

- “What tone did your partner use?”
- “Was it polite, clear, or too direct?”
  Class brief share-out — teacher highlights how tone shapes message perception.

Warm-Up (10 mins): “Tone Snapshot Poll”

- Ask the class:
  - “How often have you misunderstood tone in a message or email?”
  - “What happened?”,
  - “How did you solve it?”
- Transition: “Today, we’ll explore how to adapt tone and choose the right channel for different situations.”

I. Presentation (30 mins)

Vocabulary in Context (15 mins)
Students read three short messages in a shared doc:

1. **Email:** “Dear Alex, I wanted to follow up on the API documentation…”
2. **Slack:** “Hey, can you send the doc before stand-up?”
3. **Meeting note excerpt:** “Team agreed to merge by EOD.”

- Students highlight vocabulary words (tone, context, medium, politeness, directness) and discuss meanings in pairs.
- Teacher confirms definitions and adds examples of each in context.

Analyzing Tone (10 mins)

- Students work in groups to discuss tone (Formal ↔ Neutral ↔ Informal).
- Label each message (email, Slack, note) and explain why.
- Teacher summarizes: “Tone changes with purpose, audience, and channel.”

II. Practice (40 mins)

Choosing the Right Channel (20 mins)
Students open a shared Google Doc with 6 workplace situations:

1. Ask your team to review your code.
2. Request a day off.
3. Send a reminder about a meeting.

4. Share a company update.
5. Give feedback on someone's work.
6. Clarify a confusing ticket.

- They choose whether each should be email, chat, or call, and explain why in 1 sentence. Debrief — teacher highlights good reasoning (e.g., "email for documentation," "chat for quick coordination").

Tone Rewrite Board (20 mins)

Teacher posts 3 sample messages with problematic tone in a shared doc:

1. "Fix this now."
2. "I don't understand what you mean."
3. "You didn't update the code correctly."
   Students work in pairs to rewrite each for clarity and professionalism using the new vocabulary.
   Example rewrite:

- "Could you please take another look at this section? I think there's a small issue."
  Teacher reviews best examples with group.

[20-Minute Break]

III. Production (40 mins)

Mini-Project Simulation: "Tone Switch Challenge" (30 mins)

Students are divided into small teams. Each receives one scenario (e.g., "Manager requests a delay update" or "Client questions your code quality").

Teams must:

1. Write the same message **for two channels** — one Slack post, one formal email.
2. Adjust tone for audience and context.
3. Post both in the class Slack or shared doc.

Class gallery walk: students read other teams' posts and vote (via emojis or checkmarks) for:

- "Most clear,"
- "Most polite," and
- "Most professional."

Peer Debrief (10 mins)

Quick reflection discussion:

- "Which was easier: Slack or email?"
- "What strategies helped you keep the right tone?"

IV. Wrap-Up (15 mins)

Quick Quiz: “Tone Check”

Teacher projects 3 short messages and choose which is most appropriate for each situation. Immediate results for class discussion.

Optional Independent Practice

- Find two real professional messages (one Slack post, one email).
- Copy one example sentence from each that shows a different tone (formal vs informal)
- Write a short reflection:
    - “When would each tone be appropriate?”

      Add your reflection to the shared *Lesson 7 – Tone Practice* folder.

Notes for Instructor

- Emphasize that engineers communicate across multiple channels — tone awareness is a key soft skill.
- During tone rewrites, remind students to adjust both vocabulary and structure (not just add “please”).
- Encourage concise messages that remain polite — clarity > excessive formality.
- Keep live examples visible for constant reference (model good Slack/email posts).

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 8
Topic: Handling Disagreements and Tension
Lesson Duration: 3 hours (1hr20 – break 20 mins – 1hr20)

Can-Do Objectives (Aligned with CEFR descriptors):

- Can describe a disagreement without blaming others
- Can express concerns using softening language ("I'm wondering if…")
- Can suggest a compromise or solution diplomatically

Materials

- Slack / Discord – channel simulation for conflict messaging
- Google Docs / Notion – shared "Conflict Resolution Log" template
- Miro / FigJam – "Problem → Concern → Solution" visual board
- Mentimeter / Slido – polls and reflection

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
disagreement	a situation where people have different opinions	"We had a disagreement about deadlines."	N
conflict	serious argument or tension	"The conflict started over code-review changes."	Y
escalate	to make a conflict more serious	"Let's not escalate this issue."	Y
boundary	a limit that defines acceptable behavior	"It's okay to set boundaries on late messages."	N
compromise*	an agreement where both sides adjust	"They reached a compromise on the project timeline."	N
assertive	expressing opinions confidently but respectfully	"She was assertive without sounding rude."	N
softening language	polite phrases that reduce tension	"I'm wondering if we could try another option."	N
resolve	to find a solution or end a conflict	"The team resolved the bug priority issue."	Y

tone*	the attitude expressed in words	“His tone stayed calm even when disagreeing.”	N
empathy*	understanding another’s feelings	“Empathy helps prevent workplace conflict.”	N

Lesson Structure (PPP)

Review (10 mins): “Tone Check from Lesson 7”
Students open their previous *Tone Switch Challenge* posts.
In pairs, they identify one example where tone could cause misunderstanding.
Discuss:

- “How could you rephrase it to sound more diplomatic?”
  Teacher highlights the carry-over skill: tone awareness → conflict management.

Warm-Up (10 mins): “Disagreement Spectrum”
Ask the class:
“When your team disagrees, how do you usually react?”
Options: Avoid / Speak directly / Ask questions / Find compromise.
Quick discussion:

- “What helps you stay calm in disagreements?”
  Transition: “Today we’ll practice how to disagree diplomatically while keeping teamwork strong.”

I. Presentation (30 mins)

Vocabulary in Context

- Students read a short transcript in a shared doc:
  Liam: We should add the new feature this sprint.
  Rina: I see your point, but I’m worried it might delay other tasks.
  Liam: True. Maybe we can release a smaller version first?
  Rina: That sounds reasonable — it’s a good compromise.
- Students highlight vocabulary (softening language, compromise, tone) and infer meanings.
- Teacher confirms definitions and models pronunciation and polite intonation.

Anchored Phrases for Diplomatic Disagreement
Teacher presents a mini-list drawn from the transcript:

- “I see your point, but…”
- “I’m wondering if we could…”
- “That’s true, although…”

- “Maybe there’s a middle ground.”
  Pairs practice 2-turn exchanges using these frames in chat simulation.

II. Practice (40 mins)

Identifying the Problem and Concern

- Students open a **Miro board** divided into three columns: *Problem → Concern → Possible Solution.*
- Teacher posts 3 workplace situations (e.g., deadline conflict, code-review disagreement, feature priority debate).
- Groups drag digital sticky notes to complete the chart collaboratively.
- Debrief: each group explains one solution using softening language.

Slack Simulation: “Conflict Thread”

- In a class Slack channel (#team-discussion-practice), teacher posts:
  “@frontend wants to use a new library. @backend says it’s risky.”
  Each pair acts as one side, writing 2 messages to express concerns politely and suggest a solution.
  Example:
- “I understand the benefits, but I’m wondering if we could test it first.”
- “Good idea — maybe a small pilot would work.”
  Teacher monitors for tone, tagging strong examples.

[20-Minute Break]

III. Production (40 mins)

Mini-Project Simulation: Communication Strategy Role-Play

- Groups receive one **team-conflict scenario** (e.g., missed deadline, tech choice, meeting overload).
  They:

1. Plan a short structured conversation using the *Conflict Resolution Log* doc:
   - The issue
   - Concerns
   - Language to use
   - Proposed compromise
2. Rehearse in breakout rooms (3–5 minutes per run).
3. Perform live or record their dialogue.

After each group presents, students note one effective phrase they heard in a shared class doc.

Reflection Discussion

- “Which phrases helped keep the conversation calm?”
- “Did any language escalate tension?”

IV. Wrap-Up (15 mins)

Quick Quiz (5 mins): "Soft or Sharp?"

- Teacher reads 3 messages and students vote which is more diplomatic.
    1. "You're wrong about this." / "I'm not sure I agree — can we look at it again?"
    2. "I think there might be a better solution" / "That's not a good idea."
    3. "This isn't the most important thing right now." / "Can we circle back when this becomes more relevant?"

Reflection (10 mins):

Students complete in shared doc:

- "One phrase I'll use next time my team disagrees is…"
- "A strategy that helps me stay calm is…"

Optional Independent Practice

- Find an example of a disagreement in an online developer forum (GitHub Discussion, Stack Overflow, Reddit r/ProgrammerHumor).
- Copy one polite disagreement sentence.
- Write 3 sentences explaining why it's effective.

Notes for Instructor

- Emphasize that tone + empathy = professional conflict resolution.
- During Slack simulation, remind students to acknowledge the other side before suggesting fixes.
- Highlight cultural differences: direct vs indirect disagreement styles.
- Use group debriefs to model balanced assertiveness (neither aggressive nor passive).

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 9
Topic: Writing Clear Technical Instructions
Lesson Duration: 3 hours (1hr20 – break 20 mins – 1hr20)
Can-Do Objectives (Aligned with CEFR descriptors):

- Can write clear step-by-step instructions
- Can use sequencing and conditional phrases ("if…, then…")
- Can anticipate and address common user issues

Materials

- Whiteboard and markers (or shared slides)
- Printed or projected examples of good/bad instructions
- Vocabulary in context
- Sequence Race
- Peer Correction
- Optional: Google Docs or Notion for final writing task
- Optional: Miro / FigJam for sequencing diagram
- Small slips of paper or note cards for sorting activity

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
step-by-step	one action after another in order	"Follow these step-by-step instructions to set up the server."	Y
prerequisite	something required before starting	"Node.js is a prerequisite for running this script."	Y
sequence	the order of steps or events	"Make sure to follow the correct sequence when deploying."	Y
configuration	the setup of software or system settings	"The configuration file defines your environment variables."	Y
dependency	a program or library another program needs	"Install all dependencies before running the app."	Y
conditional	using "if…then…" logic	"If the command fails, then reinstall npm."	Y
troubleshoot	to find and fix problems	"Troubleshoot the issue by checking the error log."	Y

draft	an early version of a document	“Please submit your first draft of the installation guide.”	N
clarity*	being clear and easy to understand	“Good documentation improves clarity for users.”	N
formatting	the way text is organized and displayed	“Use consistent formatting for commands and code snippets.”	Y

Lesson Structure (PPP)

Review (10 mins): “Conflict to Clarity”

- Display a confusing or vague message on the board (e.g., “Let’s fix this quickly before the meeting”).

Instructions:

1. In pairs, students discuss:
    - What information is missing?
    - How could you rewrite it more clearly?
2. Elicit responses and write improved versions on the board.

Debrief:

Highlight how precise language and sequencing prevent miscommunication — linking this to clear technical writing.

Warm-Up (10 mins): “Bad Documentation Bingo”

Setup: Prepare 4–5 short, messy examples of poor instructions on slides or paper. Example:

“Run npm, fix the thing, it should work.”

“Click somewhere to open the file.”

Instructions:

1. In groups of three, students discuss and list all the problems they notice (vague verbs, missing order, unclear actions).
2. Each group shares one issue aloud, teacher circles common patterns on the board (e.g., unclear sequencing, missing conditionals).

Debrief:

Ask: “Why is clear writing especially important for developers and users?”

Transition to: “Today we’ll learn how to write step-by-step guides that anyone can follow.”

I. Presentation (30 mins)

Vocabulary in Context (15 mins)

Setup: Show or project a short, real-style example of clear instructions (below).

Example: Setting Up Your Development Environment

1. Install Node.js (version 18 or later).

2. Open your terminal and run `npm install`.
3. If installation fails, check your internet connection.
4. Configure your `.env` file with the correct API key.
5. Run `npm start` to launch the project.

Instructions:

1. Students read silently and underline sequencing or conditional phrases ("If installation fails, then…").
2. Pairs discuss: "What makes these instructions easy to follow?"
3. Teacher introduces new vocabulary from the text (prerequisite, configuration, troubleshoot, dependency).

Debrief:

Clarify that *technical clarity = logical order + precise verbs + readable structure.*

Phrases for Sequencing and Conditional Steps (15 mins)

Setup: Write the following on the board:

- *First / Next / Then / After that / Finally*
- *If… then… / Otherwise / Be careful to… / Don't forget to…*

Instructions:

1. Choral repetition for pronunciation and rhythm.
2. In pairs, students create a few short sentences about any real process they do at work (e.g., running a script, deploying code).
    - Example: "First, open your IDE. Then, pull the latest branch."
    - Example: "If the build fails, then check your environment variables."
3. Pairs share aloud; teacher writes the best examples on board.

Debrief:

Ask: "Which phrases help show cause and effect most clearly?"

II. Practice (40 mins)

Sequencing Race (20 mins)

Setup: Prepare a set of 8–10 instruction steps on small paper cards for each group.
Example: steps for "Making a Pull Request" or "Setting up VS Code."
Shuffle them before distributing.

Instructions:

1. Students work in small groups (3–4).
2. They must organize the steps in the correct logical order.
3. Once ordered, they read their sequence aloud using *First, Then, After that…*
4. Teacher times each group and confirms correct order.

Debrief:

Praise groups that used clear transitions and teamwork.

Peer Correction: "Fix This Documentation" (20 mins)

Setup: Project or print a poorly written short guide:

"Install packages if you need, then maybe configure settings. After, you run command. Before that, check the system works."

Instructions:

1. Students rewrite the instructions individually in clearer English.
2. In pairs, they read their new version aloud to each other.
3. Partners give verbal feedback: "That was clear because…" or "You could add a step before X."

Debrief:

Ask: "Which small words (then, after that, first) make the instructions easier to follow?"

[20-Minute Break]

III. Production (40 mins)

Mini-Project: Write a Short Technical Guide (30 mins)

Setup: Students choose a topic they know well (or teacher assigns).

Examples:

- Cloning a GitHub repository
- Installing an extension in VS Code
- Running automated tests
- Sending a Slack message with a bot

Instructions:

1. Students write 6–10 steps of clear instructions on paper or in Google Docs.
2. Encourage use of sequencing and conditionals.
3. After writing, they pair up and read their guide aloud to a partner.
4. Partner pretends to "follow" the steps verbally and asks clarifying questions.

Example exchange:

A: "First, open the terminal."

B: "Which directory should I open?"

5. Students revise unclear steps based on partner questions.

Debrief:

Ask a few students to share one "before and after" example of how they clarified a step.

Optional: Collect best examples and compile into a class "Developer How-To Library."

IV. Wrap-Up (15 mins)

Reflection Discussion (10 mins)

Instructions:

1. Students answer aloud or in writing:
    - "What made your partner's instructions clear?"
    - "What small words helped your writing flow better?"
2. Teacher elicits vocabulary and phrases for the board (clarity, sequence, conditional).

Quick Review (5 mins)

Setup: Write three short phrases on the board (e.g., “Click the button,” “If it fails, restart,” “After that, save the file”).

Instructions:

Students shout out which ones show *order* vs *condition*.

Teacher summarizes key takeaways.

Optional Independent Practice

- Find a real tech tutorial online (GitHub README, developer blog).
  Choose one section that you think is well written.
  Write 2–3 sentences explaining why it’s clear (order, word choice, tone).
  Add your reflection to the shared *Lesson 9 – Independent Practice* document.

Notes for Instructor

- This lesson focuses heavily on clarity through sequencing, keep examples short and real.
- Encourage speaking during rewriting and reviewing; avoid silent editing.
- Rotate roles: one student writes, another “follows instructions” verbally to ensure understanding.
- Allow students to draw diagrams or number steps manually if that feels more natural.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 10
Topic: Documenting Processes and Features Clearly
Lesson Duration: 3 hours (1hr20 – break 20 mins – 1hr20)
Can-Do Objectives (Aligned with CEFR descriptors)

- Describe what a feature, tool action, or process *does* using concise language
- Explain what input it needs and what the expected result is
- Write short, user-friendly documentation entries for simple processes or features

Materials

- Printed or projected *feature examples* instead of code
- Examples of short documentation (e.g., "How 'Reset Password' works")
- Paper or Google Docs for writing entries
- "Documentation Structure Frame" handout
- Partner practice activity

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
function	a feature or action that performs a specific task	"The search function finds relevant items."	Y
input	information the user or system provides	"The input is the email address."	Y
output	the result the system gives back	"The output is a confirmation message."	Y
argument	a value or option the user selects	"The filter uses date range as an argument."	Y
syntax	the standard format or style of how something is written	"Use the same syntax for all documentation entries."	Y
example	a sample showing how something works	"The documentation includes an example scenario."	Y
concise*	brief but complete	"Write a concise summary of the feature."	N
format	organize and present clearly	"Follow the same format for each entry."	Y

return	the result a function/process provides	“The return is an updated profile page.”	Y
readability	how easy text is to understand	“Good formatting improves readability.”	Y

Lesson Structure (PPP)

Review (10 mins): “Step-by-Step Recap”
From Lesson 9, show a step-by-step guide (like *Reset Password* or *Install the App*).
Ask students:

- “What was the goal of the guide?”
- “What were the main steps?”

Students summarize in 1–2 concise sentences.
Bridge:
“Today we’ll document *features* the same way—clearly and simply, using consistent structure.”

Warm-Up (10 mins): “Explain Like I’m New to the App”

- Write this on the board:
  - **“Reset Password” feature**
- Ask: “How would you explain this feature to a new user?”
- Expected student answers:
  - “It lets you create a new password if you forgot yours.”
  - “You enter your email and get a link.”
- Students work in pairs to explain:
  - What it does
  - What the user inputs
  - What the system outputs

Debrief:
Highlight *clear purpose statements* → foundation of documentation.

I. Presentation (30 mins)

Reading and Vocabulary in Context (15 mins)
Project this feature description:

Feature: Calculate Delivery Time
Purpose: Helps users estimate when their package will arrive.
Input: Location, shipping method
Output: Estimated delivery date
Example: “If you choose ‘Express,’ delivery will take 1–2 days.”

Ask students:

- What is the purpose?
- What information does the user provide (input)?

- What does the system give back (output)?
- What is the example?

Introduce vocabulary: purpose, input, output, example, concise, format.
Model:
Concise Description:
"Estimates when a package will arrive based on location and shipping type."

Phrases and Format for Documentation (15 mins)

- Write this structure:
  - **Feature Name:**
    Purpose:
    Inputs:
    Output:
    Example:
- Provide sample phrases:
  - "This feature allows users to…"
  - "The purpose of this function is to…"
  - "The user enters…"
  - "The system returns…"
  - "For example…"
- Students work in pairs to create sample documentation using the phrases provided
- Students share examples with the class

II. Practice (40 mins)

Spot the Missing Info (20 mins)
1. Display the below incomplete entry:
Feature: Convert Temperature
Description: Changes Celsius to Fahrenheit.
Tell students:
"This documentation entry is incomplete. It is missing important information a reader needs."
2. Say to students: "You will work in pairs to *complete* this entry using the documentation structure we learned."
"Add the missing parts:

- **Purpose** (Why this feature exists / what does it helps users do)
- **Inputs** (What the user provides)
- **Output** (What the feature gives back)
- **Example** (A simple illustration of how it works)

You must write a complete, clear version."
3. Give the students the structure
Write this template on the board:
Feature:
Purpose:
Inputs:

Output:
Example:
Tell students to fill in all sections.
4. Model one example quickly (not the Convert Temperature one)
    You say:
    Feature: Add to Cart
    Purpose: Lets users save an item for purchase.
    Inputs: Item selected, quantity.
    Output: Item appears in the cart.
    Example: “Add two shirts to your cart.”
Explain:
“See how each part adds clarity? Use the same logic for Convert Temperature.”
5. Students work in pairs (8–10 minutes)
Students write an improved entry for **Convert Temperature**.
    Expected student work:
    Feature: Convert Temperature
    Purpose: Helps users change a temperature from Celsius to Fahrenheit.
    Inputs: Celsius temperature.
    Output: Temperature in Fahrenheit.
    Example: “Convert 30°C → 86°F.”
6. Sharing and discussion (5–7 minutes)
- Ask 2–3 volunteers to read their improved entry.
- Guide the discussion:
    - “Is the purpose clear?”
    - “Are the inputs and outputs correct?”
    - “Is the example simple and helpful?”

Highlight 1–2 strong models on the board.

Partner Practice: Describe This Feature (20 mins)
Choose 2–3 features (no code):
- **Add to Cart**
- **Search Bar**
- **Send Verification Email**
- **Upload Profile Photo**
- **Sort by Date**

Each pair writes a documentation entry using:
- Purpose
- Inputs
- Output
- Example

Then exchange with another pair → read → give feedback:
- “This was clear because…”
- “You could add an example…”

- “The purpose could be more concise…”

III. Digital Tool (45 mins)

- To provide students the opportunity to use the digital tool in class with teacher support.
- The teacher can demonstrate activities with the whole class and/or support students as they work individually.

IV. Wrap-Up (15 mins)

Reflection Discussion (10 mins)
Students answer:

- “What helped you make your documentation clearer today?”
- “What part was hardest: purpose, input, or output?”
- “How is documenting a feature similar to writing instructions?”

Quick Recap (5 mins)
Write 3 reminders:

- Use clear structure (Purpose / Inputs / Output / Example).
- Keep sentences short and specific.
- Always think about the reader.

Optional Independent Practice

- Choose a feature from any app you use (Instagram, WhatsApp, Gmail).
- Write a documentation entry using the class format.

Notes for Instructor

- This version of the lesson removes all real code.
- “Feature documentation” mirrors “function documentation” without programming.
- Entry structure and clarity remain the focus.
- Students can apply these skills immediately at work (writing Jira tickets, user guides, onboarding docs).
- Good examples: Reset password, search bar, sort, filters, notifications, add to cart.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 11
Topic: Writing for Non-Technical Users
Lesson Duration: 3 hours (1hr20 – break 20mins – 1hr20)
Can-Do Objectives (Aligned with CEFR descriptors):

- Can rewrite technical explanations in plain English
- Can choose simpler words without changing meaning
- Can adjust tone and vocabulary for the audience

Materials

- Printed or projected technical paragraph (from real or simulated software documentation)
- Jargon match
- Vocabulary in Context
- Whiteboard and markers or shared slides
- Optional: Google Docs for collaborative rewriting
- Small slips of paper for vocabulary sorting task

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
plain English	clear, simple language for general readers	"Write in plain English so clients can understand easily."	N
simplify	make something easier to understand	"The engineer simplified the setup guide for end users."	N
jargon	specialized technical words	"Avoid jargon like 'refactor' or 'API endpoint' when writing for users."	N
term	a specific word for something	"The term 'cache' might confuse non-technical readers."	N
clarify	make something clear or less confusing	"The instructions were rewritten to clarify the steps."	N
tone*	attitude or style of writing	"Her tone was polite and reassuring."	N
explain	to describe something so others can understand it	"He explained how to reset the password in simple steps."	N
feature	a part or function of a product	"This feature allows users to export data."	Y

troubleshoot *	find and fix problems	“The support team troubleshoots login issues.”	Y
accessible	easy to understand or use	“We need to make our documentation accessible to all readers.”	N

Lesson Structure (PPP)

Review (10 minutes)

1. Write **this incomplete feature description** on the board or project it:
   Feature: Track Order
   Description: Lets users see where their package is.
2. Tell students:
   “Last lesson we practiced writing complete documentation using **Purpose – Inputs – Output – Example**.
   This entry is *not complete*. Your job is to figure out what’s missing.”
3. Students work in pairs to answer these four review questions:

Review Questions (write these on the board):

1. What is the purpose? (Why does this feature exist?)
2. What are the inputs? (What information does the user provide?)
3. What is the output? (What does the system return/give back?)
4. What is an example? (A simple “for example…” sentence)
5. After 3–4 minutes, ask pairs to write a complete version in their notebooks using the structure from last lesson:

Feature:
Purpose:
Inputs:
Output:
Example:

5. Ask one or two pairs to share their version with the class.
   Write one strong example on the board.

Warm-Up (10 mins): “Jargon Match”

Setup:

Prepare 6–8 pairs of **technical term ↔ plain English meaning** cards. Examples:

Technical Term	Plain English
API	a connection between two programs
Cache	temporary saved data
Server downtime	when the system is not working

UI	what you see and click on
Debug	find and fix a problem in code
Encryption	keeping information secure

Instructions:

1. Mix cards and distribute to students (half get “technical,” half get “plain”).
2. Students mingle to find their matching partner.
3. When all pairs are matched, they read their terms aloud to the class.

Debrief:

Ask: “Which terms are hardest to simplify?”

I. Presentation (30 mins)

Reading and Vocabulary in Context (15 mins)

Text (project or print):

Original (developer version):

“When users clear the cache, temporary data stored locally is removed, which can resolve performance issues.”

Simplified (for users):

“If your app runs slowly, try clearing saved data. This can help it work faster.”

Instructions:

1. Students read both versions silently.
2. Ask: “What changed — vocabulary or meaning?”
    - Expected: simpler verbs, shorter sentences, same message.
3. Highlight vocabulary like *cache → saved data*, *performance issues → runs slowly.*
4. Introduce and define *plain English, jargon, simplify, clarify.*

Speaking Practice:

Pairs explain what “plain English” means in their own words.

Debrief:

Summarize: “Simplifying means keeping meaning the same, not removing details.”

Tone and Audience Awareness (15 mins)

Setup:

Write three short sentences on the board:

1. “Invalid credentials detected.”
2. “Your login details are incorrect.”
3. “Hmm, that didn’t work. Please check your username and password.”

Instructions:

1. Students rank the sentences most technical → most user-friendly.
2. Ask: “Which is suitable for a system error message?” “Which fits a chatbot?”
3. Elicit reasoning (tone, formality, friendliness).

Debrief:
Highlight that tone changes depending on the product and audience.

II. Practice (40 mins)

Simplification Relay (20 mins)
Setup:
Prepare 3–4 short technical sentences on slips of paper (example below).

A. "The application will automatically synchronize user preferences across all devices."
B. "You need to initialize the configuration file before deployment."
C. "This API call returns a 404 error if the data is unavailable."

Instructions:

1. Divide students into teams of 3–4.
2. Each team receives one sentence at a time.
3. They have 1 minute to rewrite it in plain English and read it aloud.
4. The teacher or class votes if it's "clear for a non-technical user."
5. Give a new sentence until all are done.

Example Simplifications:
A → "Your settings will update automatically on all your devices."
B → "Set up the file before starting the program."
C → "If the data can't be found, the system shows an error message."
Debrief:
Encourage precision and confidence when paraphrasing.

Rewriting for Different Audiences (20 mins)
Setup:
Project one short paragraph of technical explanation:

"The backup feature creates a local copy of all stored data on your machine and uploads it to the remote server nightly via a scheduled cron job."

Instructions:

1. Students work in pairs.
2. Version 1 – rewrite for a technical colleague.
    - Include correct terms, concise phrasing.
3. Version 2 – rewrite for a client or user.
    - Use plain English, clear explanations.
4. Each pair reads both aloud; the class identifies which version matches which audience.

Debrief:
Emphasize same content, different vocabulary and tone.

[20-Minute Break]

III. Production (40 mins)

Mini-Project: Two Versions of the Same Explanation

Setup:

Provide students with 3 short technical topics (choose or assign):

1. Resetting a password
2. Syncing files between devices
3. Submitting a bug report

Instructions:

1. Each student writes two short versions of the same explanation:
    - Technical version (for developers or IT support).
    - User-friendly version (for clients or end users).
2. Students then work in pairs to read their two versions aloud to each other.
3. Partners guess which one is for technical and which for non-technical readers.
4. After guessing, pairs give feedback on word choice and tone.

Extension (optional):

Students type their "user version" into a shared class doc called *"Plain English Tech Guide."*

Debrief:

Highlight effective simplifications on the board.

IV. Wrap-Up (15 mins)

Reflection Discussion (10 mins)

Prompts:

- "What was the hardest part of simplifying your explanation?"
- "Which phrases helped make your version sound friendly and clear?"
- "Why does audience awareness matter in tech jobs?"

Instructions:

Students discuss in pairs for 3 minutes, then share one idea each.

Teacher lists key takeaways on the board:

"Short sentences, simple verbs, polite tone."

Quick Review (5 mins)

Write two pairs of words on the board:

"execute → run," "initialize → start," "synchronize → update."

Students call out which one is more user-friendly.

Optional Independent Practice

- Find one short paragraph from real documentation (e.g., AWS Docs, GitHub, Atlassian).
- Rewrite it for a **non-technical reader**.
- Write 2–3 sentences reflecting on what changed (vocabulary, tone, sentence length).
- Post in the shared *Lesson 11 – Independent Practice* doc.

Notes for Instructor

- Focus on speaking and rewriting, students should explain aloud before typing.

- Encourage creativity and humor in examples to build confidence simplifying technical text.
- Provide real examples when possible to keep relevance high.
- Reinforce that *clarity ≠ over-simplification*; goal is understanding, not dumbing down.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 12
Topic: Formatting and Visual Support
Lesson Duration: 3 hours (1hr20 – break 20 mins – 1hr20)

Can-Do Objectives (Aligned with CEFR descriptors):

- Can format a guide or article with clear visual hierarchy
- Can include screenshots, code blocks, or diagrams appropriately
- Can organize written content for readability and scanability

Materials

- Sample page from online documentation (e.g., GitHub README, Notion, Atlassian, MDN)
- Paper or shared Google Docs for student formatting task
- Optional: access to screenshot tool or diagram maker (e.g., Excalidraw, Miro)
- Printed example of well-formatted and poorly formatted text

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
heading	title or label for a section	“Use headings to divide content into sections.”	N
bullet point	a list item marker used for clarity	“Bullet points make long text easier to scan.”	N
hierarchy	organization of information by importance	“A clear visual hierarchy guides the reader’s eye.”	N
screenshot	image capture of part of a computer screen	“Add a screenshot to show where to click.”	Y
code block	formatted section for code examples	“Always place commands in a code block.”	Y
diagram	visual representation of a process or system	“Include a diagram to show how the components connect.”	Y
readability*	how easy a text is to read	“Consistent formatting improves readability.”	N
scanability	how easy it is to find key points quickly	“Use bold headings to increase scanability.”	N

caption	short text explaining a visual	“Each screenshot should have a short caption.”	N
layout	the way elements are arranged on a page	“The layout should be consistent and simple.”	N

Lesson Structure (PPP)

Review (10 mins):

1. Write the following sentence on the board or project it:

“The system has detected an authentication failure caused by invalid credentials.”

2. Tell students:
    - “Last lesson, we practiced rewriting technical language into plain English and adjusting tone for different audiences.
    - Your task now is to simplify this sentence *without changing the meaning*.”
3. Students work in pairs to answer these review questions:

Review Questions (write on the board):

1. Which words here are **jargon**?
2. How would you rewrite this for a **non-technical user**?
3. How would you rewrite it for a **chatbot**?
4. How would you rewrite it for a **developer colleague**?

Give students 3–4 minutes to write their versions in pairs.

Warm-Up (10 mins): “Find the Structure”

- Write on board:
    - to update your profile go to your account click edit fill in your info upload a photo save changes
- Students rewrite it using formatting tools:

Updating Your Profile

- Go to Account
- Click Edit
- Change your information
- Upload a photo
- Select **Save**

Brief class share.

I. Presentation (30 mins)

Vocabulary in Context (15 mins)

- Project a **real documentation snippet**

Students identify:

- headings
- subheadings

- spacing
- bullet points
- screenshots
- captions
- tables or lists

Ask:

- “Why did the writer use bullet points here?”
- “How does the layout help the reader?”

Review vocabulary in context.

Speaking: Describe the Layout (15 mins)

- Show a simple guide (like “How to change your notification settings”).
- Students describe the structure:
    - “The top heading is large.”
    - “There are two screenshots.”
    - “Each step is numbered.”
    - “The layout is clean and simple.”
- Partners ask:
    - “Why is this format effective?”
    - “Where should the visuals go?”

II. Practice (40 mins)

Formatting Fix-Up (20 mins)

- Students receive this unformatted text:
    - To create an event open the calendar choose a date click new event add details invite participants click save
- Task:
    - Rewrite it using **formatting + visuals** (students draw a tiny phone or laptop icon instead of code blocks).

Expected output:

Creating a New Calendar Event

1. Open your **Calendar**.
2. Select a date.
3. Tap **New Event**.
4. Add details.
5. Invite participants.
6. Click **Save**.

Visual Planning Activity (20 mins)

Students choose a process:

- uploading a file
- setting a profile picture
- changing a password

- adjusting notification settings
- joining a video call

On paper or Miro they plan the layout:

- heading
- list
- screenshot (draw a simple box)
- caption
- notes section

They explain their design to another group:

- “The screenshot shows where the button is.”
- “The caption explains what the user sees.”

III. Production (40 mins)

Mini-Project: Write a User Guide

Students choose a topic they know well:

- How to join a Zoom meeting
- How to create a folder on your computer
- How to send a file through Slack
- How to create a Trello card
- How to connect to Wi-Fi
- How to update your profile photo

Requirements:

- Clear title
- Sections and headings
- Bullet or numbered list
- At least one simple visual (screenshot placeholder box)
- Caption
- Optional: tips or warnings section

Partner review checklist:

✔ Strong heading
✔ Logical structure
✔ Easy to scan
✔ Helpful visual
✔ Clear layout

Students present their guide briefly.

IV. Wrap-Up (15 mins)

Reflection (10 mins)

Students answer:

- “Which formatting tool helped your writing most?”
- “What visual would you add next time?”
- “What makes a page easy to scan?”

Share 2–3 answers.

Optional Independent Practice
Students find a real online help article.
They take a screenshot and label:

- heading
- bullet points
- visual
- caption
- spacing

Then write 2–3 sentences about why the formatting works well.

Instructor's Notes:

- Emphasize that format = usability
- Use plenty of peer comparison and class discussion about what "looks clear."
- Avoid perfectionism, focus on organization and clarity, not design.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 13
Topic: Describing a System or Architecture
Lesson Duration: 3 hours (1hr20 – break 20 mins – 1hr20)
Can-Do Objectives (Aligned with CEFR descriptors):

- Can describe components of a system and how they interact
- Can use terms like *module, endpoint, service, database*
- Can explain the purpose of each part in a logical order

Materials

- Example system diagram (web app, API, or microservice architecture)
- Printed or projected technical paragraph for simplification task
- Whiteboard or digital drawing tool (e.g., Miro, Lucidchart, diagrams.net)
- Sticky notes or digital labels for system components

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
component	a part or element of a larger system	"Each component handles a specific task."	Y
module	a self-contained unit of code or functionality	"The login module manages user access."	Y
service	a function or process that provides a specific task	"The email service sends notifications."	Y
endpoint	a specific URL or interface in an API	"We connect to the `/users` endpoint to get data."	Y
database	structured data storage system	"User data is stored in a secure database."	Y
data flow	the movement of data through a system	"The diagram shows the data flow between components."	Y
backend*	the server-side part of an application	"The backend handles logic and data storage."	Y
frontend*	the part users see and interact with	"The frontend communicates with the backend via API."	Y
architecture	the structure and organization of a system	"Our system uses a microservice architecture."	Y

interaction	how components or users communicate with each other	"The service interacts with the API to retrieve data."	Y

Lesson Structure (PPP)

Review (10 mins): Jargon to Plain English

- Write this jargon sentence on the board:
  - "The system detected an issue with user authentication due to invalid credentials."
- Ask aloud:
  1. "Which words here are too technical?"
     (Expected: system, detected, authentication, invalid credentials)
  2. "What is this sentence trying to say, in simple words?"
     (Expected: you can't log in because the username/password is wrong)
  3. "How could we say this in plain English?"
     Accept 2–3 student suggestions.
- Model Answer (write on the board after discussion):
  - "Your login details are incorrect. Please try again."

Warm-Up (10 mins): "System in Your Pocket"
Instructions:

1. Ask: "What systems do you use every day?" (e.g., Spotify, Google Maps, food delivery apps).
2. Students choose one and discuss in pairs:
   - "What are the main parts?"
   - "How does it work — what talks to what?"
3. Teacher writes examples on the board:
   - "App → API → Database."
   - "User → Server → Notification Service."

Debrief:
Point out how similar structures exist across systems.

I. Presentation (30 mins)

Reading & Vocabulary in Context (15 mins)
Display this short example:

System Example: Online Food Ordering App
The frontend allows users to browse menus and place orders.
The backend processes orders and connects to the database.
The payment service communicates with an external API to confirm transactions.
Finally, the notification service updates users about delivery status.

Instructions:

1. Students read individually.
2. Ask comprehension questions:
    - "Which part interacts with the user?"
    - "What does the payment service do?"
    - "What connects to the external API?"
3. Highlight system vocabulary: *frontend, backend, service, database, API, data flow.*

Speaking Practice:

Pairs retell the system using the diagram on the board.

"The frontend sends orders to the backend, which stores them in the database."

Debrief:

Show how logical sequencing and clear verbs (send, connect, store) improve understanding.

Diagram Vocabulary and Structure Language (15 mins)

On the board or slide:

Function	Example Phrases
Introduce system	"This system consists of three main components…"
Describe parts	"The frontend handles user input."
Explain relationships	"The API communicates with the database through…"
Sequence or flow	"First, data is sent to…, then it's processed by…"
Purpose	"The main goal of this module is to…"

Instructions:

1. Review and repeat phrases aloud for pronunciation.
2. In pairs, students practice describing a simple 3-part system verbally using these patterns.

Example:

"This system consists of a frontend, a backend, and a database. The frontend sends requests to the backend, which stores the data in the database."

Debrief:

Encourage students to speak with clarity and structure, not just technical words.

II. Practice (40 mins)

Reorder the System (20 mins)

Setup:

Prepare printed slips or digital notes with parts of a system description (out of order):

- The notification service updates users.
- The database stores all order information.
- The backend processes orders.
- The user places an order in the app.

Instructions:

1. Small groups arrange the sentences into correct logical order.
2. Once finished, they draw a quick diagram to match the description.
3. Groups read their ordered version aloud.

Debrief:

Ask: “Why did you choose that sequence? What does it show about system logic?”

Diagram Walk & Talk (20 mins)

Setup:

Post 3 simple system diagrams around the room or display 3 slides:

- Example 1: Web app
- Example 2: Online shop
- Example 3: IoT home system

Instructions:

1. Divide students into groups and assign each a diagram.
2. Groups have 5 minutes to discuss and prepare a spoken explanation of how the system works.
3. Rotate groups every 5 minutes — each group explains a new diagram to their classmates.

Teacher Tip: Encourage use of sequencing phrases (*first, next, then, finally*) and interaction verbs (*sends, connects, communicates, retrieves*).

Debrief:

Highlight clarity and accuracy in verbal explanations.

[20-Minute Break]

III. Production (40 mins)

Mini-Project: System Diagram and Explanation

Instructions:

1. Students choose one system they know (personal project, product, or company tool).
2. They create a simple architecture diagram with 3–6 components (can be hand-drawn).
3. They write a short, structured explanation (6–8 sentences):
    - What are the main components?
    - How do they interact?
    - What is the purpose of each?
4. In pairs, students present their diagram orally (2–3 minutes each).
    - Partner listens and asks one clarifying question: “Where does the data go after that?”
5. Presentations may be done on paper, whiteboard, or slide.

Peer Checklist:

- Clear component names
- Logical flow (input → process → output)
- Correct technical vocabulary

- Uses sequencing language
- Easy to follow visually

Debrief:

Praise structured explanations and smooth presentation flow.

IV. Wrap-Up (15 mins)

Reflection Discussion (10 mins)

Prompts (on board):

- "What helped make your explanation clear?"
- "What was hard to describe visually?"
- "How can you simplify technical information for non-developers?"

Pairs discuss; volunteers share with class.

Teacher summarizes:

"Clear system descriptions show logic, flow, and purpose — not just complexity."

Quick Recap (5 mins):

Students shout out one useful phrase from today ("This system consists of…", "The backend processes…").

Optional Independent Practice

- Find an architecture diagram online (e.g., AWS, Firebase, or GitHub project README).
- Write a short (5–6 sentence) paragraph describing how it works.
- Include sequencing and interaction language.

Notes for Instructor

- Encourage speaking over drawing perfection — the goal is explanation clarity.
- For less visual learners, provide pre-made diagrams for description practice.
- Model several ways to introduce systems ("This system has three layers…").
- Use consistent correction on structure and flow words rather than technical detail.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 14
Topic: Simplifying Complex Technical Ideas for Non-Technical Audiences
Lesson Duration: 3 hours (1hr20 – break 20 mins – 1hr20)
Can-Do Objectives (Aligned with CEFR descriptors)
By the end of this lesson, students can:

- Explain complex or technical ideas using clear, simple language
- Use analogies and examples to make explanations clearer
- Check comprehension and restate ideas using paraphrasing

Materials

- Handout: "Plain English Phrase Bank"
- Short text: "Technical vs Plain English Example"
- Jargon Cards (API, cloud, debugging, latency, encryption, repository)
- Student worksheets (simplification tasks, rewriting tasks, analogy prompts)

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
simplify*	make something easier to understand	"We need to simplify this explanation for the client."	N
clarify*	make something clearer	"Let me clarify what the backend does."	N
audience	the people you are explaining to	"Adjust your explanation based on your audience."	N
avoid jargon	not use specialized language	"Try to avoid jargon with non-technical listeners."	N
plain English*	clear and simple language	"Explain the feature in plain English."	N
analogy	comparing something to a familiar idea	"An analogy helps explain complex systems."	N
paraphrase	restate something using different words	"Let me paraphrase that more simply."	N
assumption	something you expect the listener to know	"Don't assume the audience knows technical terms."	N

example*	a simple case that shows the idea	“Give an example to make your point clearer.”	N
explanation	describing how something works or why	“Your explanation was clear and easy to follow.”	N

Lesson Structure (PPP)

Review: Fix the system sentence (10 mins)

1. Write one short, incorrect system sentence on the board:
   “The app sends a notification before the user orders food.”
2. Tell students:
   “This sentence is not correct. As a class, let’s fix it.”
3. Ask the students these two questions:
   “What is wrong here?”
   (The app does *not* send a notification before the user orders.)
   “What should happen first?”
   (The user should open the app → then choose food → then get updates.)
4. Let students raise hands and suggest the correct order.
   You don’t need any technical knowledge — just guide the order logically.

Example sentence to write on the board:
“First the user orders food in the app.
Then the app sends a notification about the order.”

Warm-Up (10 mins): “Explain to a 10-Year-Old”

- Give students Jargon Cards (API, cloud, debugging, latency, encryption, repo).
- Students explain each in one simple sentence to a partner.
- Volunteers share their simplest explanations.

Debrief
Highlight: a good analogy or comparison makes complex ideas accessible.

Vocabulary Introduction (10 mins)

1. Show the vocabulary table.
2. Students listen and repeat pronunciation.
3. Project vocabulary matching worksheet on the board without the terms:

A. “Let me say it another way so it’s easier to understand.”
B. “Who am I talking to? A client? An engineer?”
C. “In simple terms, it’s like a digital lock.”
D. “Don’t use too many technical terms here.”
E. “Let me make this clearer.”
F. “A short story or comparison that helps understanding.”
G. “I thought you already knew what a database was.”
H. “This is a simple version of the idea.”

I. “This sentence needs more clear details.”
J. “Explaining something in different words.”

4. Students match each sentence to the vocabulary word.
5. Partner check.

Transition
“Now let’s see how simplifying language looks in real technical writing.”

I. Presentation (30 mins)

Reading (15 mins): Technical vs Plain English
Text
Technical version:
“Our distributed microservice architecture handles asynchronous event processing with fault tolerance.”
Plain English version:
“Our system is made of small parts that work independently. They handle tasks at the same time and keep working even if one part fails.”
Teacher Steps

- Students read both versions.
- Ask:
  - “Which is easier to understand?”
  - “How did the writer simplify the explanation?”
- Underline: short sentences, familiar words, order, analogies, no jargon.

Phrase Bank (15 mins)
Students receive handout with phrases:
Introducing an explanation:

- “In simple terms…”
- “What this really means is…”

Clarifying:

- “Another way to say this is…”
- “Let me rephrase that…”

Checking understanding:

- “Does that make sense?”
- “Should I explain that part again?”

Practice
Students use 2–3 phrases to explain a simple technical idea (e.g., “How search works,” “What a bug is”).

II. Practice (40 mins)

Jargon Replacement Challenge (20 mins)
Students rewrite short, jargon-heavy sentences into plain English.

Example items:

- "We need to optimize latency."
- "The API returned a 403 error."
- "The repository needs refactoring."
- "Deploy the updated container."

Pairs compare answers and choose the clearest versions.

Rewriting for a Non-Technical Audience (20 mins)

Students rewrite product update notes for three audiences:

- a PM
- a customer
- a friend who is not in tech

Example:

"Implemented client-side caching to reduce server load during peak hours."

Students produce three audience-specific versions.

III. Production (40 mins)

Mini-Project: Explain a Technical Idea Three Ways

Student Steps

Choose a topic:

- how login works
- how an app loads
- what cloud storage is
- what debugging means
- how notifications work

Explain the same idea in **three versions**:

1. For a *technical* audience
2. For a *semi-technical* audience
3. For a *non-technical* audience

Each version should include:

- 3–5 sentences
- 2 plain-English phrases
- 1 analogy
- 1 comprehension check ("Does that make sense?")

Peer Review

Partner asks 1–2 follow-up questions.

IV. Wrap-Up (15 mins)

Reflection (10 mins)

Students write responses to:

- "Which phrase today helped you explain things more clearly?"
- "What was hardest to simplify?"
- "Where do you need plain English at work?"

Quick Recap Game (5 mins): “Simplify This!”
Teacher calls out technical terms; students simplify them quickly (10 seconds each).
Terms: repository, server, encryption, deployment, latency, debugging, API.

Optional Independent Practice
Write two versions of a technical concept from your job:

- one jargon-heavy technical explanation
- one plain-English explanation

Upload to the class folder.

Instructor Notes

- Emphasize clarity, not correctness.
- Encourage analogies and real-life comparisons.
- Correct students when they “simplify” by adding *different* jargon.
- Celebrate clear explanations — clarity is a professional skill in high-tech roles.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 15
Topic: Explaining Relationships and Dependencies
Lesson Duration: 3 hours (1hr20 – break 20 mins – 1hr20)
Can-Do Objectives (Aligned with CEFR descriptors)

- Use conditional structures like *"If X happens, then Y will…"*
- Explain basic dependencies in systems, workflows, or tasks
- Clearly explain cause, effect, and consequences

Materials

- Simple dependency chain diagram (human workflow + system workflow)
- Short text: "What Happens If…" workplace scenario
- Conditional phrase bank handout
- Scenario cards (non-technical and semi-technical)
- Dependency chain worksheet

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
dependency*	something that must happen before something else can	"Task B has a dependency on Task A."	N
relationship	how two things are connected	"There's a clear relationship between time and workload."	N
conditional*	an if–then rule	"If the user forgets their password, then they must reset it."	Y
sequence*	the order events happen in	"There's a specific sequence for releasing updates."	Y
trigger	something that starts a process	"Publishing a post triggers a notification."	Y
impact*	the effect of a change or problem	"A delay in design impacts the whole sprint."	N
cause	the reason something happens	"Low bandwidth caused the video to buffer."	N
effect	the result of a cause	"The effect was slower load times."	N

scenario	a possible situation	“In this scenario, the app won’t load.”	N
consequence	what happens because of an action	“A missed deadline has serious consequences.”	N

* review words

Lesson Structure (PPP)

Review (10 mins):

1. Write ONE technical sentence on the board:
   “The server is experiencing high latency.”
2. Say:
   “Last class, we practiced turning jargon into plain English and explaining things simply.
   Let’s rewrite this sentence so a 10-year-old could understand it.”
3. Ask the class these prompts (students answer aloud):

Write the guided prompts on the board or ask aloud:

1. **Which words are jargon?**
   (Students should identify: *server*, *latency*.)
2. **What does this really mean?**
   (Expected ideas: “It’s slow,” “It’s taking a long time.”)
3. **How can we say this in simple English?**
   Accept 3–4 student suggestions.
4. **Can someone add a simple analogy?**
   Example prompt:
   “What is something in real life that slows down? A car? A line at the store?”
   Students generate analogies:
   - “The computer is moving slowly, like waiting in a long line.”
   - “The system is stuck in traffic.”

Warm-Up (10 mins): “Chain Reactions” (Simplified Version)
Instructions

Give each student a sticky note. Students write a common workplace step, such as:

- Write requirements
- Create design
- Code
- Review
- Test
- Merge
- Release
- Announce to users

Students stand in a line, forming a “workflow chain.”

Teacher calls out an event:

- "Design is unclear."
- "Testing takes longer than expected."
- "Code review is blocked."

Each student reacts with one sentence:

- "If design is unclear, then development slows down."
- "If review is blocked, testing can't start."

I. Presentation (30 mins)

Vocabulary Introduction (10 mins)

1. Project the vocabulary table.
2. Students match example sentences with vocabulary terms by themselves or in pairs

Transition

"Now that you know the key terms, let's look at how they appear in real workplace scenarios."

Language Focus: Conditionals in Real Workplace Context (15 mins)

Teacher Guide: Understanding Conditionals (Zero, First, Second)

Zero Conditional

Form: *If + present simple, present simple*

Use: *General truths, rules, things that are always true*

Examples:

- *"If you heat water, it boils."*
- *"If the Wi-Fi is down, messages don't send."*

Teacher Tip:

Zero conditional = ***facts****.*

You can replace "if" with "whenever" and the meaning stays the same.

First Conditional

Form: *If + present simple, will + base verb*

Use: *Real, possible situations in the future*

Examples:

- *"If the designer finishes today, we will start development tomorrow."*
- *"If the server restarts, the app will work again."*

Teacher Tip:

First conditional = ***real future consequence****.*

If this happens → then that will happen.

Second Conditional

Form: *If + past simple, would + base verb*

Use: *Imaginary, unlikely, or hypothetical situations (NOT real now)*

Examples:

- *"If we had more time, we would improve the UI."*
- *"If the server were faster, the system would load more quickly."*

Teacher Tip:
Second conditional = ***imagination****.*
Not happening now — just a possibility in theory.

Quick Summary

Conditional	***Meaning***	***Example***	***Memory Trick***
Zero	*Always true*	*"If it rains, the ground gets wet."*	*facts / rules*
First	*Real future*	*"If it rains, we'll stay inside."*	*possible future*
Second	*Imaginary*	*"If it rained right now, we'd cancel the game."*	*hypothetical*

- Explain to the students that there are 3 different kinds of conditionals. Provide the above table without technical topics for them to first understand the difference of when you use each conditional and the grammar differences.
- Ask students to create their own conditional sentences first using conditional zero, then once they have a firm grasp, conditional 1 and then 2.
- Once students have a better understanding of conditionals in general, move onto an example related to computer science:

Type	**Example**	**Meaning**
Zero	"If a feature is not tested, it doesn't get released."	General rule
First	"If the designer finishes today, we can start development tomorrow."	Real future
Second	"If the team had more time, the quality would be better."	Imaginary / unlikely

Student Task

- Students create one sentence for each type using workplace tasks (not deep technical systems).

Examples:

- "If the requirements change, we update the plan."
- "If QA finds an issue, we will fix it."

Reading: "What Happens If…" Workflow Example (15 mins)

Text

A typical team workflow has four steps: planning, design, development, and testing.
If planning is incomplete, the design team cannot start.
If design is late, development slows down.

If development is blocked, testing will not begin on time.
If testing finds major issues, the release is delayed.

Task

- Students underline all if–then sentences.
- Comprehension questions:
    - "What happens if planning is incomplete?"
    - "What is the effect of late design?"
    - "Why might a release be delayed?"

Pairs rewrite one sentence using synonyms:

- "In case…"
- "Whenever…"
- "If this part is delayed…"

II. Practice (40 mins)

Dependency Mapping (20 mins)

- Give students a simple *workplace* dependency chain:
    - **Requirements → Design → Development → Testing → Release**

Student instructions

1. Identify 3–4 dependencies in the chain.
2. Write conditional sentences for each:
    - "If design takes longer than expected, development will be delayed."
    - "If testing finds issues, release will be postponed."
3. Share with the class.

Scenario Cards (20 mins)

- A teammate is sick and cannot finish their part.
- The design file has the wrong version.
- Testing took longer than expected.
- The client requests a change mid-sprint.
- The internet connection is unstable during deployment.
- A meeting is canceled last minute.

Student task

Create two sentences:

1. If–then result
2. Possible solution

Example:

- "If the designer uses the wrong file, then development will start with incorrect information."
- "We should confirm the file version before starting."

Pairs present and discuss.

III. Digital Tool (45 mins)

- To provide students the opportunity to use the digital tool in class with teacher support.
- The teacher can demonstrate activities with the whole class and/or support students as they work individually.

IV. Wrap-Up (15 mins)

Reflection (10 mins)
Ask students:

- "Which conditional sentence today felt most natural?"
- "Where do you see dependencies in your everyday work?"
- "How can conditional language make communication clearer?"

Optional Independent Practice

- Write 4 sentences describing a dependency in your work:
- 1 cause → effect sentence
- 1 effect → consequence sentence
- 1 conditional sentence
- 1 solution sentence

Instructor Notes

- This lesson avoids complex architecture or backend systems — focus is workplace logic, not technical depth.
- Encourage students to use natural workplace examples, not strict engineering ones.
- Correct conditional structures gently; focus on *clarity over grammar*.
- Prompt students who struggle by asking:
    - "What depends on what?"
    - "What happens if…?"
    - "What is the consequence?"

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 16
Topic: Justifying Technical Decisions
Lesson Duration: 3 hours (1hr20 – break 20 mins – 1hr20)
Can-Do Objectives (Aligned with CEFR descriptors):

- Can explain why one solution was chosen over another
- Can compare options using evaluative and comparative language ("more scalable," "less secure," "easier to maintain")
- Can support arguments with examples or expected outcomes

Materials

- Example system diagrams (e.g., monolithic vs microservice architecture)
- Handout or slide: comparison vocabulary and justification phrases
- Whiteboard or slides for pros/cons brainstorm
- Paper or digital space for mini-project diagrams
- Optional: Miro or Lucidchart for visual creation

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
scalability*	ability to handle growth efficiently	"Microservices are more scalable than monoliths."	Y
maintainability *	how easy it is to update or fix a system	"We chose this framework for better maintainability."	Y
trade-off*	a balance between two desirable but conflicting features	"We made a trade-off between speed and security."	Y
reliability*	ability to perform consistently under conditions	"The new server setup increases reliability."	Y
performance*	how fast or efficiently a system runs	"Caching improves performance by reducing load time."	Y
security	protection against unauthorized access or attacks	"This API uses encryption for better security."	Y

cost-effective	providing good value for money	“The cloud solution is more cost-effective than hosting locally.”	N
flexible	easily adaptable to change	“This architecture is flexible for future features.”	N
drawback	disadvantage or limitation	“One drawback of this tool is its limited community support.”	N
reasoning	the explanation behind a choice or belief	“Clear reasoning builds credibility in technical discussions.”	N

Lesson Structure (PPP)

Review (10 mins): Conditionals
Ask the class these questions:
1. “What is wrong with this sentence?”
   (Expected answer: Planning happens *before* development, so planning cannot be delayed by development.)
2. “What should happen first in a real workflow?”
   (Expected: Planning → Design → Development → Testing → Release)
3. “So what is the correct conditional?”
   Guide students toward something like:
   Corrected Sentence:
   “If planning is delayed, development will start late.”

Write the final answer on the board.

Student Quick Practice (pair work, 2–3 minutes)
Tell students:
“Take 30 seconds to choose ANY two steps in a workflow and create your own conditional sentence.”
Example student answers:
- “If design is unclear, development slows down.”
- “If testing takes longer than expected, release will be delayed.”
- “If requirements change, the plan will need an update.”

Take 1–2 volunteers to share.

Warm-Up (10 mins): “Pros and Cons Wall”
Setup:
Write two columns on the board: **Advantages** | **Disadvantages**.
Instructions:

1. Give students a topic: *"Using cloud infrastructure."*
2. Pairs brainstorm 2–3 pros and cons.
3. Write examples on the board.
    - Pros: scalable, easy to deploy
    - Cons: recurring cost, security concerns
4. Discuss which points are strongest.

Debrief:
Highlight words that express evaluation: *more/less, easier/harder, better/worse.*

I. Presentation (30 mins)

Input Reading: Example Comparison (15 mins)

Text:

We chose a microservice architecture instead of a monolith.
The main reason was scalability — each service can grow independently.
Although this setup increases complexity, it improves performance and deployment speed.
Another advantage is reliability: if one service fails, the others still function.
The drawback is higher maintenance effort, but the trade-off is worth it for large-scale systems.

Instructions:

1. Students read silently.
2. Ask:
    - "Why did the team choose microservices?"
    - "What was the trade-off?"
    - "How did they justify their choice?"
3. Highlight comparative and reasoning words: *because, although, but, the main reason, advantage, drawback, trade-off.*

Debrief:
Explain that justification combines **comparison + reasoning**.

Phrase Bank: Explaining and Justifying Choices (15 mins)

Function	Example Phrases
Giving reasons	"We chose this approach because…" / "The main reason was…"
Showing trade-offs	"Although this adds complexity, it improves scalability."
Comparing	"This solution is more flexible and easier to maintain."
Speculating on results	"This change should reduce response time."

Conceding	“It’s more expensive, but it’s also more reliable.”

Instructions:

1. Review each phrase aloud and write example sentences together.
2. In pairs, students complete prompts orally:
    - “We decided to use ____ because…”
    - “Although it’s ____, it helps us ____.”
3. Collect best examples and write them on the board.

Debrief:

Point out strong logic connectors (*because, although, but, therefore*).

II. Practice (40 mins)

Decision Comparison Task (20 mins)

Setup:

Display three short project scenarios:

A. Building an internal tool for 10 users.

B. Designing a mobile app for 1 million users.

C. Setting up a data storage system for analytics.

Instructions:

1. Pairs discuss which technologies or architectures they would choose (monolith vs microservices, SQL vs NoSQL, etc.).
2. Each pair writes 3–4 sentences explaining their choice using target phrases.
   Example:

   “We’d choose SQL because it’s more structured and easier to manage.”
   “Although NoSQL is faster, it’s less reliable for financial data.”

3. Pairs share one answer each with the class.

Debrief:

Highlight how reasoning changes with context (small vs large scale).

Ranking Debate (20 mins)

Setup:

Write on the board: *performance, cost, scalability, security, maintainability.*

Instructions:

1. In small groups, students rank these factors by importance for a new project.
2. Groups discuss and justify their order:
    - “We put performance first because users care about speed.”
    - “Security is more important for banking apps.”
3. Each group presents their top 3 priorities to the class.

Debrief:

Discuss how engineers must justify *why* they prioritize certain aspects.

[20-Minute Break]

III. Production (40 mins)

Mini-Project: System Diagram Walkthrough and Justification
Instructions:

1. Each student selects a real or fictional system they know (e.g., ticketing app, online store, IoT network).
2. Prepare a short (3–5 minute) presentation including:
    - A **diagram** showing components and flow
    - An **explanation** of how it works
    - **One or more design choices** with justification (e.g., "We used NoSQL for scalability.")
3. Include one **visual aid** (diagram or slide).
4. After presenting, students answer **1–2 peer questions**.

Peer Checklist:

- Clear description of system components
- Logical explanation of relationships/dependencies
- At least one justified design decision
- Confident delivery and clear visuals

Debrief:
Applaud confident explanations, clear logic, and professional phrasing.

IV. Wrap-Up (15 mins)

Reflection Discussion (10 mins)
Prompts:

- "What phrases helped you justify your decisions?"
- "Which trade-offs are common in your work?"
- "How can explaining your reasoning improve teamwork?"

Pairs discuss; volunteers share one takeaway.

Optional Independent Practice
Find a public tech article (e.g., Medium, Dev.to) where someone explains a design decision.
Write 3–4 sentences summarizing:

- What choice they made
- Why they made it
- What trade-off was involved

Post your reflection in the *Lesson 16 – Independent Practice* doc.

Notes for Instructor

- Keep focus on spoken justification and structured reasoning.
- Encourage use of visuals — even simple diagrams reinforce explanation clarity.

- Push for variety in connectors (*although, while, even though, therefore, however*).
- Remind students that justifying decisions is a key professional skill in meetings and documentation.

Course: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: 17
Topic: Sprint Planning and Task Assignment
Lesson Duration: 3 hours (1hr20 – break 20 mins – 1hr20)
Learning Outcomes

- I can describe work tasks and estimate time clearly
- I can negotiate responsibilities politely
- I can suggest and agree on priorities using appropriate phrases

Materials

- Short meeting transcript (for reading)
- Whiteboard or projector
- Printed backlog list (for sorting task)
- Role cards (Project Manager, Developer, Tester, Designer)
- Mini-project sheet (for group simulation)
- Peer checklist
- Reflection prompts slide or board

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
backlog	list of all tasks waiting to be done	"Let's review the backlog before we start the sprint."	Y
sprint	short work period for completing goals	"We plan tasks for the next sprint every Monday."	Y
assign	give a task to someone	"I assigned the bug fix to Dana."	Y
estimate	guess the time or effort needed	"We estimated this task will take two days."	Y
priority	task that's more important than others	"Fixing login errors is our top priority."	N
negotiate	discuss to reach an agreement	"We negotiated who would take each task."	N
dependency *	when one task depends on another	"Testing depends on finishing the backend first."	Y
deadline	final date to finish work	"Our deadline is next Friday."	N

capacity	amount of work a team can complete in one sprint	"Our team's capacity is about 40 hours per sprint."	Y
blocker	a problem that stops work	"A missing API key was the main blocker last sprint."	Y

Teacher Context (for non-technical instructors)

*A **sprint** is a short work cycle in software teams (usually 1–2 weeks) where the team decides which tasks to complete.*
A sprint planning meeting is when the team:

- *Reviews the list of upcoming work (called the backlog)*
- *Chooses tasks for the sprint*
- *Decides who will do what*
- *Sets priorities and time estimates*

The goal of this lesson is to help students discuss work tasks politely and efficiently, using real workplace English (e.g., "Can you take this one?", "That makes sense.").

Lesson Structure (PPP)

Review: Giving Reasons (10 minutes)
Write this sentence on the board:
Incorrect Justification Sentence:
"We chose a monolith because it is more scalable than microservices."
Tell students:
"Last lesson, we practiced giving reasons, using trade-off language, and choosing correct comparisons.
This sentence has incorrect reasoning. Let's fix it together."
Class Discussion (Teacher asks, students answer aloud)
1. "What is wrong with this sentence?"
Expected answers:
- Monoliths are usually *less* scalable than microservices.
- The justification does not match common reasoning.

2. "What would be a *true* reason to choose a monolith?"
Students may respond:
- "It's easier to maintain for small teams."
- "It's simpler and needs less infrastructure."
- "It's cheaper and faster to build."

3. "Let's rewrite the sentence correctly."
Guide students to produce something like:
Corrected Version (write on board):
"We chose a monolith because it is simpler to maintain for our small team."

or
“Although microservices are more scalable, a monolith is easier for us to build and manage.”

Warm-Up (10 mins): “Team Discussion”

1. Teacher Asks: “When you work in a team, how do you decide who does what?”
2. Write sentence starters on the board:
    - “I usually volunteer for…”
    - “We decide tasks based on…”
3. Students discuss for 3–4 minutes in pairs.
4. Gather a few ideas and transition:
   “In software teams, this is exactly what happens in sprint planning.”

I. Presentation (30 mins)

Short Reading: “Team Meeting Transcript”
Teacher Steps:

1. Read the text aloud or play it as a short dialogue (two students can read roles).
2. Ask:
    - “Who is doing what?”
    - “What did they decide to delay?”
3. Write the polite phrases on the board:
    - “I can take…”
    - “Can we…?”
    - “That makes sense.”

Text:

PM: Let’s check the backlog. We still need to fix the login error and finish the dashboard.
Dev 1: I can take the dashboard, but I’ll need two more days.
QA: That’s fine. Can we delay the API update until next sprint?
PM: Yes, that makes sense.

Phrase Focus Practice

Function	Example Phrases
Suggesting	“Can we…?”, “What if we…?”, “Maybe we should…”
Agreeing	“That makes sense.”, “I agree.”
Negotiating	“I can…, but I’ll need…”, “Could you take…?”

Teacher Steps:

1. Students work in pairs to create one new example for each category.
2. Example:

- *Suggesting:* “Can we finish the bug fix first?”
- *Negotiating:* “I can do that if someone helps with testing.”

3. Share 2–3 examples with the class.

II. Practice (40 mins)

Task Sorting Challenge (20 mins)

What Students Do:

Decide which tasks should go into a short sprint and justify their decisions.

Teacher Steps:

1. Show this simple backlog:

Task	Time (hours)
Fix login bug	2
Build profile page	5
Write release notes	1
Test analytics	3
Deploy update	2

2. In groups of 3, tell students:
   “You have 10 hours available. Choose which tasks fit and explain why.”
3. Encourage polite negotiation using target phrases:
   “Can we move this task to next sprint?” / “That makes sense, it’s low priority.”
4. Each group shares one sentence summarizing their plan.

Expected Output:

“We’ll do the login bug, testing, and deployment because they’re most urgent.”

Sprint Planning Role-Play (20 mins)

1. Assign roles (simple labels):
   - *Project Manager, Developer, Tester, Designer*
2. Write a scenario on the board:
   “Your team has one week to finish a website. You must decide: design, test, deploy.”
3. Students talk for 5–6 minutes and decide who will do what.
4. Circulate, listening for use of negotiation phrases.

What Students Do:

Practice giving and discussing short task assignments.

Expected Output:

“Can we start with the design?” / “I can test the form pages.” / “That makes sense, let’s deploy at the end.”

III. Production (40 mins)

Mini Sprint Simulation

Teacher Steps:

1. Put students in teams of 4–5.
2. Write a simple project on the board (e.g., *"Create a company website"*).
3. Each group:
    - Lists 5–7 tasks.
    - Chooses which 3–4 fit in the sprint.
    - Decides who will do each task.
4. Each group gives a short 2-minute summary of their plan.

What Students Do:

Work in small groups to plan and present a short sprint.

Expected Output:

"We'll focus on the login page, testing, and deployment. I'll take the testing, and Dan will deploy."

Peer Checklist:

- Clear, polite language
- Logical priorities
- Balanced task assignments

IV. Wrap-Up (15 mins)

Reflection Discussion (10 mins)

Prompts (on board):

- "Which phrases helped your group collaborate?"
- "Was it easy or difficult to decide tasks?"
- "What's one new word you'll use in your next meeting?"

Pairs discuss briefly; then share with the group.

Optional Independent Practice

- Students find an online backlog example (from GitHub, Jira, or a template).
- They write 5–6 sentences describing what the team should focus on and why.

Notes for Instructor

- Don't worry about real Agile rules — focus on clear English for teamwork and planning.
- Encourage tone, clarity, and teamwork, not technical accuracy.
- Remind students to keep answers short and natural.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 18
Topic: Running and Participating in Retrospectives
Lesson Duration: 3 hours (1 hr 20 – break 20 mins – 1 hr 20)
Can-Do Objectives (Aligned with CEFR descriptors):

- I can give positive and constructive feedback
- I can suggest process changes using diplomatic language
- I can summarize group reflections clearly

Materials

- Sample "Retrospective Notes" reading text
- Phrase-bank handout for giving feedback and suggestions
- Whiteboard or projector
- Team reflection worksheet
- Mini-project reflection board (digital or paper)
- Peer feedback checklist

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
retrospective	meeting to reflect after a project or sprint	"We had a team retrospective to discuss what went well."	Y
feedback*	comments to help improve performance	"I gave feedback on the testing process."	N
constructive*	helpful and positive in tone	"Her feedback was constructive and specific."	N
improvement	change that makes something better	"We suggested improvements to our communication."	N
reflection	thinking carefully about past actions	"The team shared reflections on the sprint."	N
suggestion	an idea for improvement	"He made a suggestion to simplify the workflow."	N
tone*	the attitude or feeling in communication	"We should keep a positive tone during feedback."	N
outcome*	the result of an activity	"The outcome of our sprint was a stable release."	Y

progress	movement toward goals	“We made good progress this sprint.”	N
challenge	something difficult that must be overcome	“Time management was our biggest challenge.”	N

Teacher Context (for Non-Technical Instructors)
In Agile software teams, a retrospective is a short meeting at the end of a sprint where the team reflects on their work.
They discuss three key questions:

1. *What went well?*
2. *What could be improved?*
3. *What will we do differently next time?*

This lesson helps students practice giving feedback and discussing improvements politely and constructively, which are useful skills for any professional meeting, not just in software development.

Review (10 mins): Reflecting on Progress
Goal: Connect previous sprint planning and stand-up lessons to reflection.
Teacher Steps:

1. Ask: “After finishing a big project, what do you usually talk about with your team?”
2. Write students’ ideas on the board under *What went well* and *What could be better*.
3. Summarize: “This kind of meeting is called a retrospective.”

Expected Output:
Students recognize that reflection is a normal part of improving teamwork.

Warm-Up (10 mins): “Positive First”
Goal: Set a constructive tone for giving feedback.
Teacher Steps:

1. Write two sentences on the board:
   - “Your report was too long.”
   - “Your report was very detailed. Next time, maybe make it shorter.”
2. Ask: “Which sentence sounds more polite?”
3. Discuss: Why is the second one more effective?
4. Elicit useful diplomatic words: *maybe, perhaps, we could, next time.*

Expected Output:
Students understand how to turn criticism into constructive feedback.

I. Presentation (30 mins)

Reading: Sample Retrospective Notes (15 mins)
Text:

What went well: The new testing tool saved us time. Communication between QA and Developers improved.

What could be improved: We didn't have enough time for design reviews. Some tasks were not clearly assigned.

Action items: Schedule design review meetings earlier. Define responsibilities more clearly next sprint.

Teacher Steps:

1. Read or project the text.
2. Ask comprehension questions:
    - "What went well?"
    - "What problems did they notice?"
    - "What will they do next time?"
3. Highlight phrases that show a positive tone:
    - "We could improve…"
    - "Let's make sure…"
    - "It might help if…"

Expected Output:

Students identify the 3-part structure and polite phrasing.

Useful Phrases (15 mins)

Function	Example Phrases
Praising or recognizing	"I think we did a great job with…" / "I really liked how we handled…"
Identifying problems	"One challenge was…" / "We had some difficulty with…"
Suggesting improvements	"Maybe we could…" / "It might help if we…" / "Let's try to…"
Summarizing feedback	"So, the main takeaway is…" / "Next time we should focus on…"

Teacher Steps:

1. Go over each phrase category.
2. Ask students to match one phrase with a line from the text.
3. Model short exchanges:
   A: "One challenge was communication."
   B: "Right, maybe we could plan shorter meetings next time."

II. Practice (40 mins)

Team Reflection Worksheet (20 mins)

Goal: Practice identifying successes and challenges.

Teacher Steps:

1. Distribute the following table or project it on the board:

What went well	What could be improved	Suggestions for next sprint

2. Students complete the chart in pairs, thinking about a recent project, assignment, or even class activity.
3. Encourage positive, specific examples.
4. Example answers:
    - *What went well:* "Teamwork was strong."
    - *Could be improved:* "Time management."
    - *Suggestions:* "Start tasks earlier."

Pair Feedback Exchange (20 mins)
Goal: Practice giving and responding to feedback politely.
Teacher Steps:

1. Write prompts on the board:
    - "I liked how you…"
    - "One thing we could improve is…"
    - "Maybe next time we could…"
2. Students take turns giving feedback to their partner about a past group task or class activity.
3. Partners respond positively (e.g., "Thanks, that's a good point." / "I agree, let's try that.").

Expected Output:
Short 2–3 line exchanges with polite tone and clear suggestions.

III. Production (40 mins)

Mini-Project – Team Retrospective Simulation
Goal: Simulate a real sprint retrospective using a visual board.
Teacher Steps:

1. Groups of 4–5 create a retrospective board with three sections:
    - What went well
    - What could be improved
    - Ideas for next sprint
2. Give each group a simple project scenario (e.g., "Designing a company website," "Creating an internal app").
3. Students write 2–3 notes per column, then discuss them as a team.
4. Each group presents a short 2-minute summary.

Peer Checklist:

- Positive and constructive tone
- Clear suggestions for improvement

- Organized and professional summary

Expected Output:

"We worked well as a team, but we could improve communication. Next sprint, we'll start daily check-ins."

IV. Wrap-Up (15 mins)

Reflection (10 mins)

Prompts:

- "What makes feedback helpful instead of negative?"
- "How can you keep a positive tone?"
- "Which new phrase will you use next time you give feedback?"

Optional Independent Practice

Write a short retrospective report (5–6 sentences) about your last group project or work week. Include one positive comment, one challenge, and one suggestion for improvement.

Notes for Instructor

- Remind students to balance positives and suggestions.
- Focus on tone and clarity, not technical accuracy.
- Encourage every student to contribute at least one idea in discussions.
- Optional: use sticky notes or a shared digital board (Miro, Jamboard, etc.) to make it interactive.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 19
Topic: Facilitating Collaboration and Decision-Making
Lesson Duration: 3 hours (1 hr 20 – break 20 mins – 1 hr 20)
Can-Do Objectives (Aligned with CEFR descriptors):

- Can lead or facilitate discussions to reach a decision
- Can summarize and confirm group decisions clearly
- Can involve quiet members and manage time effectively

Materials

- Short meeting transcript (for reading)
- Phrase-bank handout for leading discussions
- Timer for meeting facilitation activity
- Group task cards with decisions to make
- Mini-project brief for group decision-making scenario
- Peer checklist for evaluating facilitation

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
facilitate	to guide or manage a discussion	"She facilitated the team meeting efficiently."	N
decision	a conclusion or choice after discussion	"We made the decision to delay the release."	N
agenda	list of items to discuss	"The agenda includes planning and testing updates."	N
summarize*	to repeat the main ideas in short form	"Let me summarize what we agreed on."	N
clarify*	to make something clear or easy to understand	"Can I clarify what you mean by 'deadline'?"	N
feedback*	comments about performance or results	"The facilitator asked for feedback after the meeting."	N
suggestion*	an idea or proposal for improvement	"That's a good suggestion — let's discuss it."	N
participation*	being involved and active in a discussion	"Good facilitation encourages everyone's participation."	N

consensus*	a decision everyone agrees on	“The team reached consensus to delay the feature.”	Y
time limit	a fixed amount of time for a task	“Let’s set a 10-minute time limit for each topic.”	N

Teacher Context (for Non-Technical Instructors)
In professional teams (including software teams), someone often takes the role of facilitator — a person who guides a meeting so everyone contributes, the group stays on topic, and a decision is made.
A facilitator doesn’t need to be the boss; their job is to:

- *Open the discussion clearly*
- *Keep the group focused*
- *Encourage quieter people to speak*
- *Summarize decisions before ending*

The goal of this lesson is to help students lead short, structured discussions politely and confidently, using clear meeting language.

Review (10 mins): What Makes a Good Meeting?
Goal: Activate prior knowledge and connect to professional meeting skills.
Teacher Steps:

1. Write on the board: *productive, respectful, clear, short*.
2. Ask: “What makes a meeting successful?”
3. Elicit ideas (e.g., clear agenda, equal speaking time).
4. Summarize: “Today we’ll practice how to **lead** these kinds of meetings.”

Expected Output:
Students share ideas like “Stay on topic,” “Include everyone,” “End with decisions.”

Warm-Up (10 mins): “The Bad Meeting”
Goal: Identify poor meeting habits through humor.
Teacher Steps:

1. Read this short “bad meeting” dialogue aloud:

   Alex: We need to decide on the logo today.
   Dana: Wait, can we talk about next month’s event?
   Sam: I don’t think this is my job.
   Alex: Okay, we’re out of time. Bye.

2. Ask: “What went wrong?” (Elicit: no agenda, no focus, no summary.)
3. Write key points: *Agenda, Focus, Summary, Participation*.

Expected Output:
Students identify problems and agree a facilitator could help.

I. Presentation (30 mins)

Reading: Short Meeting Transcript (15 mins)

Text:

Facilitator: Let's start with our main goal — choosing a testing tool. We have 15 minutes.
QA: I suggest Tool A because it's faster.
Developer: Tool B is easier to integrate.
Facilitator: Okay, good points. Let's take one more comment before deciding.
Designer: I agree with QA; speed matters more.
Facilitator: Great, so we'll go with Tool A. Any objections? No? Perfect — next step is to schedule the setup.

Teacher Steps:

1. Read or play the dialogue aloud.
2. Ask: "What did the facilitator do well?" (Elicit: time management, balance, summarizing.)
3. Highlight phrases used for leading discussion:
    - "Let's start with…"
    - "We have 15 minutes."
    - "Let's take one more comment."
    - "Any objections?"
    - "Perfect — next step is…"

Expected Output:

Students can identify 3–4 expressions for guiding discussion.

Useful Phrases (15 mins)

Function	Example Phrases
Starting the meeting	"Let's begin with…", "Our goal today is…"
Managing time	"We have 10 minutes for this topic.", "Let's stay focused on…"
Encouraging participation	"What do you think, <name>?", "Does anyone else want to add something?"
Summarizing decisions	"So, we agreed to…", "The next step is…"
Closing politely	"Thanks everyone — great input today.", "Let's follow up next time."

Teacher Steps:

1. Go over each category with examples.
2. Model short facilitator–team exchanges:

    A: "Let's begin with our first topic — testing."
    B: "I think we should start with Tool A."

A: “So, we agree on Tool A. Next step: setup.”

3. Practice pronunciation and tone (confident, polite).

II. Practice (40 mins)
The Meeting Maze (20 mins)
Goal: Practice guiding short 5-minute group discussions.
Teacher Steps:

1. Give each group a **Meeting Task Card** with one decision:
    - Choose a team name
    - Decide which feature to fix first
    - Plan the next team event
2. Assign one student as **Facilitator**.
3. Facilitator leads for 5 minutes using the phrase list.
4. Groups summarize their decision in one sentence.
5. Rotate facilitators for a new topic.

Expected Output:

“We decided on the name CodeWave because it’s simple.”

Summarizing the Decision (20 mins)
Goal: Practice clear closing statements.
Teacher Steps:

1. Write this structure on the board:
    - “We discussed…”
    - “We agreed to…”
    - “Next step is…”
2. Each group writes and presents a short summary (1–2 sentences).
3. Give feedback on clarity and tone.

Expected Output:

“We discussed two options and agreed to use Tool A. Next step is to test it tomorrow.”

III. Production (40 mins): Mini-Project – Lead a Team Decision Meeting
Goal: Students facilitate a short meeting to solve a realistic team problem.
Teacher Steps:

1. Divide class into teams of 4–5.
2. Give each team a **Scenario Card** (examples below).
3. Each group selects one student as the facilitator.
4. The facilitator runs a 7-minute meeting while others participate.
5. Each team summarizes their final decision for the class.

Scenario Examples:

- The team must choose between two tools for testing.
- The team needs to decide how to improve communication.

- The team must plan the next sprint's focus.

Peer Checklist:

- The facilitator guided discussion politely
- Everyone had a chance to speak
- The meeting ended with a clear summary

Expected Output:

> "We agreed to improve communication by using daily check-ins. Everyone will post updates by noon."

IV. Wrap-Up (15 mins)

Reflection (10 mins)

Give these discussion topics to the class, they can discuss as a class or in small groups

- "What makes a good facilitator?"
- "What was hardest about leading?"
- "Which phrases helped you manage the meeting?"

Optional Independent Practice

Write a short summary of a meeting you attended (real or imagined).

Include:

- The goal of the meeting
- One or two ideas discussed
- The final decision or action item

Example:

> The goal of the meeting was to plan the next sprint. We discussed three priorities and decided to focus on bug fixes.

Notes for Instructor

- Encourage confident but neutral tone when leading.
- Remind facilitators to include quieter members by name.
- Time each group meeting to keep pace and realism.
- Praise positive leadership and organization.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 20
Topic: Agile Tools and Documentation
Lesson Duration: 3 hours (1 hr 20 – break 20 mins – 1 hr 20)
Can-Do Objectives (Aligned with CEFR descriptors):

- Can explain how tasks are tracked and updated
- Can describe how information is documented and shared
- Can respond to tool-related questions or issues

Materials

- Sample Jira or Notion board screenshot (fictional or printed)
- Short reading: "Team Update in Jira"
- Phrase-bank handout for tool descriptions and issue responses
- Whiteboard or projector
- Digital task card examples (Trello, Confluence, Notion, etc.)
- Peer checklist for final presentations

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
tool	software used to perform a task	"We use Jira as our main project management tool."	Y
track	to follow progress or changes	"We track all open bugs in Jira."	Y
update	to change information to reflect progress	"I updated the task status to 'Done'."	Y
assign*	to give a task to someone	"The manager assigned the task to me."	Y
comment	to add feedback or notes	"I left a comment on the issue for QA."	Y
document	to record information formally	"We document sprint goals in Confluence."	Y
workflow	the process or sequence of work	"Our workflow includes code review and testing."	Y
dashboard	screen showing project data	"The dashboard displays current progress."	Y

repository	a digital storage location for files or code	“We keep all our designs in a shared repository.”	Y
status	the current stage of a task	“The task status changed from ‘To Do’ to ‘In Progress’.”	Y

Teacher Context (for Non-Technical Instructors)
*Agile teams use **digital tools** (like Jira, Confluence, Trello, or Notion) to track tasks, store documentation, and share updates.*
This lesson helps students practice describing how they use such tools in English — for example:

- *Explaining where tasks are tracked (“We use Jira to manage our backlog”)*
- *Talking about progress (“I updated the task status to ‘In Review’”)*
- *Explaining documentation (“We record our meeting notes in Confluence”)*

*Teachers do **not need to know how to use these tools**, the goal is to help students describe and explain their digital workflows clearly and professionally.*

Lesson Structure (PPP)
Review (10 mins): Common Agile Tools
Goal: Connect students’ real-life experience with the lesson topic.
Teacher Steps:

1. Write on the board: *Jira – Trello – Notion – Confluence – Slack.*
2. Ask: “Which tools do you use for work?” (If none, ask: “Which tools do you know?”)
3. For each tool, elicit its purpose in one simple sentence:
    - “Jira → tracking tasks”
    - “Confluence → storing documentation”
    - “Slack → quick team communication”
4. Write examples on the board for reference.

Expected Output:
Students describe the purpose of common Agile tools in plain English.

Warm-Up (10 mins): “Show and Tell – Tools Edition”
Goal: Encourage students to describe familiar tools using simple language.
Teacher Steps:

1. Ask each student to think of one tool they use at work or school.
2. In pairs, they answer:
    - “What do you use it for?”
    - “How often do you use it?”
    - “What do you like or dislike about it?”
3. Circulate and help with phrasing (e.g., “We use it to…”).

Expected Output:

"We use Notion to organize our tasks." / "I use Google Docs to share notes with my team."

I. Presentation (30 mins)

Reading: "Team Update in Jira" (15 mins)

Text:

We use Jira to organize our sprint tasks. Each task has a title, description, and status. Developers move their tasks from "To Do" to "In Progress" and then "Done." We also add comments to explain blockers or progress. This helps everyone see what's happening in real time.

Teacher Steps:

1. Read the text aloud or have one student read.
2. Ask comprehension questions:
    - "What does the team use Jira for?"
    - "What do they do when a task is finished?"
    - "How do they communicate blockers?"
3. Highlight useful verbs: *organize, move, add, see.*

Expected Output:

Students can describe how teams use digital boards to track work.

Useful Phrases (15 mins)

Function	Example Phrases
Describing a tool	"We use [tool] to manage our projects." / "It helps us track progress."
Talking about updates	"I moved the task to 'Done'." / "I added a comment to explain the issue."
Describing documentation	"We document meeting notes in [tool]." / "The wiki includes all team guidelines."
Responding to tool issues	"I can't find that task — maybe it's under a different status." / "Can you update the link?"

1. Read the phrases aloud and clarify meaning.
2. Students repeat in pairs, replacing [tool] with their own examples.
3. Model a short conversation:

    A: "Where can I find the design notes?"
    B: "They're documented in Confluence."

II. Practice (40 mins)

Describe Your Workflow (20 mins)
Goal: Practice explaining how tasks are tracked and documented.
Teacher Steps:

1. Provide a simple fictional task board (printed or slide):

Task	Status	Tool
Fix login bug	In Progress	Jira
Update homepage content	Done	Confluence
Test mobile app	To Do	Trello

2. In pairs, students explain the board using target language.
3. Example:
   "We track our bugs in Jira. The login bug is still in progress. Content updates are already documented in Confluence."
4. Encourage use of connectors: *also, first, next, finally.*

"Tool Troubleshooting" Role-Play (20 mins)
Goal: Practice responding to tool-related questions or problems.
Teacher Steps:

1. Give each pair a problem card (examples below).
2. Students take turns asking for and giving help using polite, practical language.

Problem Cards:

- "You can't find your assigned task."
- "Someone forgot to update a task to 'Done.'"
- "A link in the documentation is broken."
- "Two people are editing the same document."

Example Dialogue:

A: "I can't find my task on the board."
B: "Maybe it's under a different status. Try checking 'In Progress.'"

Expected Output:
Students use tool vocabulary and problem-solving language naturally.

III. Digital Tool (45 mins)

- To provide students the opportunity to use the digital tool in class with teacher support.
- The teacher can demonstrate activities with the whole class and/or support students as they work individually.

IV. Wrap-Up (15 mins)

Reflection (10 mins)

Prompts:

- "Which tool is most useful for your work or studies?"
- "What phrases can you use to explain how your team tracks progress?"
- "How can you describe problems politely?"

Quick Recap (5 mins)

Teacher calls out: *Describe, Update, Respond.*

Students give quick examples:

"Describe — We use Jira to track bugs."

"Update — I moved the task to Done."

"Respond — Can you update the link?"

Optional Independent Practice

Write a short paragraph (5–6 sentences) about the tools your team uses. Include:

- What each tool is for
- How often it's updated
- Why it's useful

Example:

Our team uses Jira for tracking and Notion for documentation. We update tasks every morning. It helps everyone see what's finished and what's next.

Notes for Instructor

- Emphasize everyday English, not technical tool use.
- Allow students to choose familiar tools if possible.
- If learners don't use Agile tools, use school or project examples (e.g., "We use Google Sheets to track homework").
- Keep focus on clarity and real-world communication.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 21
Topic: Describing and Reproducing Complex Bugs
Lesson Duration: 3 hours (1 hr 20 – break 20 mins – 1 hr 20)
Can-Do Objectives (Aligned with CEFR descriptors):

- Can describe an issue with context, steps, and outcomes
- Can use technical language like “stack trace,” “intermittent,” or “reproduce”
- Can summarize what makes a bug complex

Materials

- Short “bug report” reading text
- Phrase-bank handout for describing technical problems
- Whiteboard or projector
- Printed or digital “bug report form” worksheet
- Simplified log or screenshot sample
- Peer checklist for short presentations

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
bug	an error that causes a program to behave incorrectly	“We found a bug in the login feature.”	Y
reproduce	to make a problem happen again	“Can you reproduce the error on your computer?”	Y
intermittent	happening sometimes, not always	“It’s an intermittent issue — hard to catch.”	Y
stack trace	a list of program actions showing where an error occurred	“The stack trace shows the failure in line 45.”	Y
symptom	a visible sign of a problem	“One symptom is that the screen freezes.”	N
environment	the system or setup where software runs	“The bug only appears in the test environment.”	Y
root cause	the main reason a problem occurs	“The root cause was a missing configuration file.”	Y

workaround	a temporary solution before fixing a bug	“We created a quick workaround to avoid crashes.”	Y
log	recorded details of actions or errors	“Check the error log for more information.”	Y
regression	when a fix causes an old problem to return	“The update introduced a regression in user login.”	Y

Teacher Context (for Non-Technical Instructors)
A bug is a problem in software that causes something to behave incorrectly.
When engineers describe a bug, they usually explain:

1. *What happened (symptoms)*
2. *Where it happened (the environment or feature)*
3. *How to reproduce it (steps that cause it again)*
4. *Why it’s complex (intermittent, difficult to trace, or caused by multiple factors)*

This lesson helps students describe problems clearly and professionally — useful for writing tickets, reporting issues, or explaining challenges in meetings.
Teachers do not need to understand code; the focus is on clear English for problem explanation.

Review (10 mins): What Is a Bug?
Goal: Activate prior knowledge and ensure understanding of key vocabulary.
Teacher Steps:

1. Write on the board: *bug, reproduce, environment, fix.*
2. Ask: “What do you think each word means in your work or studies?”
3. Elicit short examples and clarify definitions in plain English.
4. Example:

> “A bug is like a mistake in the code. To reproduce it means to make it happen again.”

Expected Output:
Students use basic technical terms accurately and confidently.

Warm-Up (10 mins): “The Mystery Problem”
Goal: Practice describing symptoms clearly.
Teacher Steps:

1. Display or read this example aloud:

> “Sometimes the login page shows a white screen after clicking ‘Submit,’ but it doesn’t

happen every time."

2. Ask: "What do you know about the problem from this description?"
3. Discuss: Why is it difficult to fix? (Elicit: it's intermittent, no clear pattern.)
4. Optional: Have students create a simple non-technical example ("The printer works for one person but not another").

Expected Output:
Students can describe a problem and identify why it's tricky.

I. Presentation (30 mins)

Reading: Example Bug Report (15 mins)

Text:

Title: Login Page Crashes Intermittently
Description: The login page sometimes freezes after clicking "Submit." It happens in Chrome and Safari on macOS.
Steps to Reproduce:

1. Open the login page.
2. Enter valid credentials.
3. Click "Submit."
   Expected Result: User logs in successfully.
   Actual Result: Screen becomes blank and unresponsive.
   Additional Info: Stack trace shows a missing API response.

Teacher Steps:

1. Read the text together or display it on screen.
2. Ask comprehension questions:
   - "What is the bug?"
   - "How often does it happen?"
   - "What is the cause?"
3. Highlight key phrases:
   - "It happens in…"
   - "Steps to reproduce…"
   - "Expected vs. actual result."

Expected Output:
Students understand structure and tone of a professional bug report.

Useful Phrases (15 mins)

Function	Example Phrases
Describing the issue	"The app crashes when I click…" / "It only happens on…"
Giving context	"This issue appears in the latest version of…" / "It works locally but fails on the server."

Explaining reproduction	“To reproduce the bug, follow these steps…” / “I can reproduce it consistently.”
Explaining complexity	“It’s intermittent.” / “The root cause seems unclear.” / “The error log shows…”

Teacher Steps:

1. Review each phrase with students.
2. Practice pronunciation of key terms (*intermittent, reproduce, root cause*).
3. In pairs, students use the phrases to describe an imaginary bug.

Example Dialogue:

A: “The app crashes when I click Save.”
B: “Can you reproduce it?”
A: “Yes, it happens every time on mobile.”

II. Practice (40 mins)

“Describe the Bug” (20 mins)
Goal: Practice describing technical issues in detail.
Teacher Steps:

1. Give students one of the following **Bug Scenario Cards** (or display on slides):
   - The website loads slowly only on mobile.
   - The app crashes when uploading large files.
   - The dark mode doesn’t apply to pop-up windows.
   - The notification sound plays twice.
2. Students write a 3–4 sentence description including:
   - Where the issue happens
   - Steps to reproduce
   - Expected vs. actual result
3. Pairs exchange and ask clarifying questions.

Expected Output:

“The app crashes when uploading files larger than 10MB. It happens in Chrome. To reproduce, upload any big file. The app closes unexpectedly.”

“Find the Cause” (20 mins)
Goal: Encourage logical reasoning and teamwork language.
Teacher Steps:
Give each group a short *bug log sample* (provided in materials).
Example Log Snippet:

```
Error: API Timeout at line 232
Stack trace: user_service.js → getUserData()
```

1.

2. Ask: “What might this mean?” (Guide discussion: “Maybe the server took too long to respond.”)
3. Each group writes one short explanation of the likely cause and one possible fix.

Expected Output:

“The root cause might be a slow API response. A workaround could be adding a timeout retry.”

III. Production (40 mins): Mini-Project – Bug Report Simulation

Goal: Create and present a professional bug report.

Teacher Steps:

1. Divide students into small teams (3–4).
2. Give each team one new scenario (or let them invent one).
3. Each team creates a short bug report with:
    - Title and description
    - Steps to reproduce
    - Expected vs. actual results
    - Possible cause or next step
4. Teams present their report in 2–3 minutes.

Peer Checklist:

- Clear structure and language
- Accurate use of vocabulary
- Logical description of problem and result

Expected Output:

“Our report is about a bug in the dashboard filters. The issue happens when selecting multiple options. The page doesn’t refresh correctly. The possible cause is a missing event trigger.”

IV. Wrap-Up (15 mins)

Reflection (10 mins)

Prompts:

- “What makes a good bug report?”
- “Why is clarity more important than technical detail?”
- “Which new words will help you describe problems at work?”

Quick Recap (5 mins)

Teacher calls: *Describe*, *Reproduce*, or *Root Cause.*

Students respond with a short sentence.

“Describe — The screen freezes after clicking Submit.”

“Reproduce — Follow the same steps, and it happens again.”

Optional Independent Practice

Write your own bug report (5–6 sentences) about a real or imaginary issue.
Include:

- Title
- Description
- Steps to reproduce
- Expected vs. actual result

Example:

Title: Chat window doesn't scroll automatically
Description: When new messages arrive, the chat window doesn't scroll to the latest message. Happens in Firefox only.
Steps to Reproduce: Send three messages in a row.
Expected Result: Chat should scroll automatically.
Actual Result: User must scroll manually.

Notes for Instructor

- Focus on language accuracy and logical structure, not real debugging.
- Clarify terms as needed using analogies (e.g., *log = error diary*).
- Encourage teamwork and active questioning ("Can you reproduce it?").
- Allow humor and creativity to keep engagement high.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 22
Topic: Explaining Root Causes
Lesson Duration: 3 hours (1 hr 20 – break 20 mins – 1 hr 20)
Can-Do Objectives (Aligned with CEFR descriptors):
Can explain the causes of problems using technical language
Can describe the chain of events that led to an error
Can respond to follow-up questions about the cause
Materials

- Short reading: "Root Cause Analysis Example"
- Cause-and-effect diagram handout ("fishbone" or "5 Whys")
- Phrase-bank for explaining causes and reasoning
- Printed "Problem Chain" cards
- Mini-project worksheet for team presentations

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
root cause*	the main reason a problem happens	"The root cause was a missing configuration file."	Y
chain of events	connected actions leading to a result	"The chain of events started with a wrong API call."	N
error log	a file showing program failures	"The error log helped identify the issue."	Y
trigger*	an action that starts a process or problem	"Saving the file triggered the crash."	Y
malfunction	when something stops working properly	"A sensor malfunction caused incorrect data."	N
hypothesis	a theory or educated guess	"Our hypothesis was that the cache was full."	N
correlation	connection between two things	"There's a correlation between network load and crashes."	N
misconfiguration	wrong system settings	"A misconfiguration in the server caused delays."	Y

dependency*	one part needing another to work	“The feature failed because of a missing dependency.”	Y
fix / patch	a solution to an error or bug	“We released a patch to fix the root cause.”	Y

Teacher Context (for Non-Technical Instructors)
*A **root cause** is the main underlying reason a problem happens.*
In software, engineers ask “Why?” several times to trace an issue back to its source.
Example:

- *The app crashed → Why? Memory overflow → Why? A loop never stopped → Why? Missing exit condition.**

*This lesson helps students practice describing **why** something failed and **how** they discovered it using logical sequencing, cause–effect connectors, and clear professional English.*

Review (10 mins): What Causes Problems?
Goal: Activate cause–effect thinking and link to previous lesson.
Teacher Steps:

1. Write on board: *bug → crash → restart → delay.*
2. Ask: “What caused what?”
3. Chain together with student help: “The bug caused the crash, which led to a restart and a delay.”
4. Highlight connectors: *because, so, therefore, as a result.*

Expected Output:
Students build simple cause–effect sentences correctly.

Warm-Up (10 mins): “The 5 Whys Challenge”
Goal: Introduce root-cause questioning.
Teacher Steps:

1. Write a light, non-technical example on board: *The student was late to class.*
2. Ask “Why?” five times, writing each answer: traffic → accident → rain → no umbrella → left home late.
3. Explain this is how engineers find root causes.
4. In pairs, students do their own 5-Whys for a funny scenario (e.g., “The coffee machine overflowed”).

Expected Output:
Logical 5-Why chains spoken aloud with humor.

I. Presentation (30 mins)

Reading: Root Cause Analysis Example (15 mins)
Text:

Incident: Website went offline for 20 minutes. \n> **Immediate Cause:** Database connection failed. \n> **Root Cause:** Expired SSL certificate blocked access. \n> **Resolution:** Renewed certificate and added monitoring alert.

Teacher Steps:

1. Read aloud or project.
2. Ask: “What was the problem?” “What caused it?” “What fixed it?”
3. Highlight clear sequencing and short, factual style.

Expected Output:

Students identify cause, root cause, and solution.

Useful Phrases (15 mins)

Function	Example Phrases
Describing the cause	“The issue was caused by …”, “It happened because …”
Showing sequence	“First …, then …, after that …”
Explaining analysis	“We checked the logs and found …”, “Our investigation showed …”
Giving results	“As a result …, so …, therefore …”
Answering questions	“Yes, we tested it again after the fix.”, “We confirmed it wasn’t a network issue.”

Teacher Steps:

1. Read each set aloud.
2. Have pairs create one mini dialogue using 2–3 phrases.
3. Example:

A: “The error happened because the database was full.”
B: “So the root cause was a storage limit?” A: “Exactly.”

II. Practice (40 mins)

Cause-and-Effect Diagram (20 mins)

Goal: Visualize how problems develop.

Teacher Steps:

1. Hand out the “fishbone” diagram or draw it on the board.
2. Write one central problem: *Website loading slowly.*
3. Label branches: Server / Network / Database / Code / User Actions.
4. Groups brainstorm one possible cause for each branch and present their diagram.

Expected Output:

“Server – low memory. Network – high traffic. Database – slow query.”

Problem Chain Cards (20 mins)
Goal: Use sequencing language to describe how a problem unfolds.
Teacher Steps:

1. Give each pair a set of Problem Chain Cards (shuffled).
   Example cards: "User uploads image" → "File not validated" → "Server crashes" → "Site offline."
2. Students arrange the cards in logical order and retell the story using connectors.
3. After telling, swap sets with another pair.

Expected Output:

"First, the user uploaded a large file. Then the file wasn't validated, so the server crashed and the site went offline."

III. Production (40 mins): Mini-Project – Explain a Root Cause

Goal: Present a root-cause explanation for a fictional incident.
Teacher Steps:

1. Groups of 3–4 choose one scenario (or create their own):
   - *System slowdown after update*
   - *User data missing after import*
   - *Payment system error at checkout*
2. Groups prepare a 2-minute presentation explaining:
   - What happened
   - The chain of events
   - The root cause
   - The solution or next step
3. Encourage visuals (fishbone, timeline, simple chart).

Peer Checklist:

- Clear description of problem and root cause
- Logical sequence of events
- Professional language and tone
- Team participation

Expected Output:

"The payment error was caused by an expired API key. It started after the latest update. We renewed the key and added a monitor to prevent it."

IV. Wrap-Up (15 mins)

Reflection (10 mins)
Prompts:

- "What's the difference between a symptom and a root cause?"
- "How can you explain a complex issue clearly to non-technical people?"
- "Which phrases helped you sound professional?"

Quick Recap (5 mins)

Teacher calls out a connector (*because, so, therefore, as a result*).
Students create a quick sentence about a bug using that word.

Optional Independent Practice

- Write a short paragraph (6–7 sentences) explaining a problem and its root cause.
- Include: what happened, why it happened, and how it was fixed.

Example:

> "The website was slow because the database index was missing. We checked the logs and found many slow queries. After adding an index, the speed improved by 40%."

Instructor Notes:

- Use simple analogies for technical concepts (e.g., *root cause = main reason*).
- Encourage students to speak logically and use connectors for flow.
- Remind them to differentiate between symptom and cause.
- Praise clear, structured explanations over technical accuracy.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 23
Topic: Optimization and Improvements
Lesson Duration: 3 hours (1hr20 – break 20 mins – 1hr20)
Can-Do Objectives (Aligned with CEFR descriptors):

- Can describe changes that improved performance or reliability
- Can compare performance before and after a fix
- Can use terms like “load time,” “latency,” “resource usage”

Materials

- Short text: “Performance Optimization Example”
- Comparison chart template (Before vs. After)
- Phrase bank for describing improvements
- Improvement cards (examples of optimizations)
- Mini-project handout and peer checklist
- Timer or stopwatch for speaking activities

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
optimization	process of making something as effective as possible	“We optimized the app to load faster.”	Y
latency	delay between action and response	“We reduced latency by 30%.”	Y
performance *	how fast or efficiently something works	“Performance improved after the update.”	Y
resource usage	how much memory or CPU a process consumes	“We lowered resource usage by 15%.”	Y
load time	time it takes for a page or app to open	“The new version cut load time in half.”	Y
scalability*	ability to handle more users or data	“The new system is more scalable.”	Y
cache	temporary storage that improves speed	“We added a cache to reduce API calls.”	Y
throughput	amount of data processed in a given time	“The server’s throughput doubled.”	Y

reliability*	consistency of performance without failure	“Reliability improved after fixing the error.”	Y
trade-off*	balance between two competing factors	“We improved speed but used more memory — a trade-off.”	Y

Lesson Structure (PPP)

Review (10 mins): Connecting to Root Causes
Teacher Steps:

1. Write on the board: *problem → cause → improvement*.
2. Ask students to share one example from their work or a past lesson.
3. Highlight that optimization means *moving from cause → solution → improvement.*

Expected Output:

“The bug caused crashes. We fixed the API call. Performance improved by 40%.”

Warm-Up (10 mins): “Before and After”
Teacher Steps:

1. Write simple examples:
    - *Before: Slow website (8 sec load)*
    - *After: Fast website (2 sec load)*
2. Ask: “How can we describe the improvement?”
3. Students respond: “The load time was reduced from 8 seconds to 2 seconds.”
4. In pairs, students create 1–2 of their own “Before and After” examples (can be tech-related or daily life).

Expected Output:

“Before: My laptop took 5 minutes to start. After: It starts in 30 seconds.”

I. Presentation (30 mins)

Reading: Performance Optimization Example (15 mins)
Text:

Scenario: Our web application was taking too long to load the dashboard.
Analysis: The main issue was large images and unnecessary API calls.
Action: We compressed images and implemented data caching.
Result: Load time decreased from 6 seconds to 2 seconds, and user satisfaction improved.

Comprehension Questions:

1. What was the problem?
   What caused it?
   What actions were taken?
2. What improved as a result?

Answer Key:

1. Dashboard loaded slowly.
2. Large images and repeated API calls.
3. Compressed images and added caching.
4. Load time and user satisfaction improved.

Useful Phrases: Describing Improvements (15 mins)

Function	Example Phrases
Describing a change	"We improved… by…", "We implemented… to…", "We reduced…"
Making comparisons	"Before, it took…, but now it takes…", "It's faster / more stable / more efficient than before."
Quantifying results	"Speed increased by 40%.", "Memory usage dropped to half."
Explaining impact	"This helped users access data faster.", "The system became more reliable."
Discussing trade-offs	"It's faster now, but it uses slightly more memory."

Teacher Steps:

1. Review phrases aloud.
2. Have students match each phrase to an example from the reading text.
3. Pairs practice short mini-dialogues describing their own imagined improvements.

II. Practice (40 mins)

Comparison Chart (20 mins)

Goal: Practice describing performance changes using data.

Teacher Steps:

1. Provide this chart to pairs.
2. Students fill it out with improvements, either from the text or their own experience.

Feature	Before	After	Improvement
Page load time	6 seconds	2 seconds	Reduced by 4 seconds
API calls	10 per user	3 per user	Simplified process
Server CPU usage	90%	60%	More efficient

Error rate	5%	1%	More reliable

3. Have pairs explain their chart aloud using connectors (“before,” “after,” “as a result”).

Expected Output:

“Before the update, the error rate was 5%, but after optimization, it dropped to 1%.”

Optimization Cards (20 mins)

- Distribute or project one optimization scenario per group.

Cards:

1. The mobile app used too much battery. You changed how data syncs.
2. The homepage loaded too slowly. You added caching.
3. The API server was overloaded. You balanced traffic.
4. The images were too large. You compressed them.
5. The code ran in a loop. You added a break condition.

Task:

Groups describe the improvement using the structure:

- Problem → Cause → Action → Result.

Example Output:

“The app used too much battery because of background sync. We changed the sync interval, and battery life improved by 30%.”

III. Production (40 mins)

Mini-Project – Optimization Report

Goal: Practice clear, data-driven explanation of improvements.

Teacher Steps:

1. Groups of 3–4 select a real or imaginary optimization project (system update, code refactor, UI redesign).
2. Prepare a 2–3 minute presentation with:
    - The problem
    - What caused it
    - What was done
    - The result (use imaginary metrics)

Example:

“Our website used to take 8 seconds to load. We reduced image size and implemented lazy loading. Load time decreased to 2 seconds, and bounce rate dropped by 25%.”

Peer Checklist:

Criteria	Yes / No
Explained problem and improvement clearly	☐
Used data or comparisons	☐

Used correct technical terms	☐
Spoke fluently and clearly	☐

IV. Wrap-Up (15 mins)

Reflection Prompts:

- What's the best optimization you've ever done or heard of?
- Which phrases today help describe performance data clearly?
- How can you describe results without sounding too technical?

Quick Recap Game:

Teacher calls out: *speed*, *memory*, *reliability*, *latency*.
Students create one-sentence improvements for each.

"We improved latency by 40%." / "The system is more reliable after caching."

Optional Independent Practice

Write a short paragraph (6–8 sentences) about an improvement you made (real or fictional).
Include:

- The problem
- The cause
- The change made
- The result

Example:

"We improved the mobile app's speed by optimizing image size and reducing API calls. The average load time dropped from 5 to 2 seconds, and user satisfaction increased."

Instructor Notes:

- Correct tone and clarity, not technical accuracy.
- Encourage concise explanations: 1–2 sentences per step.
- Push students to use numbers (even if estimates).
- Provide praise for strong logical structure.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 24
Topic: Writing and Presenting Technical Fixes
Lesson Duration: 3 hours (1hr20 – break 20 mins – 1hr20)
Can-Do Objectives (Aligned with CEFR descriptors):

- Can write a structured report or ticket comment describing a fix
- Can explain steps, tools, and reasoning clearly
- Can answer clarifying questions from teammates

Materials

- Short reading: Example Fix Report
- Phrase bank for fix descriptions and explanations
- Before/After log snippets
- Fix Report template
- Q&A prompt cards
- Observer checklist for presentations

Vocabulary
(*asterisk = previously introduced word*)

Term	Definition	Example Sentence	Profession-Specific (Y/N)
fix*	a solution to a technical problem	"We deployed a fix for the caching issue."	Y
patch	a small update that repairs a bug	"The patch resolved the timeout error."	Y
regression*	a bug that reappears after a change	"This regression appeared after the new release."	Y
reproduce*	to trigger a bug again	"We reproduced the issue with a test script."	Y
investigate	to examine and understand the cause	"We investigated and found a missing dependency."	N
root cause*	the underlying reason for a problem	"The root cause was a broken API call."	Y
apply a fix	to put the change into action	"We applied a fix to the validation function."	Y
verify	to check whether something works	"We verified the fix in staging."	Y

log output	text showing system behavior	“The log output showed a 500 error.”	Y
rollout	the process of releasing a fix	“The rollout was completed with zero downtime.”	Y

Review (10 mins)
Bug → Cause → Fix Chain (Pairs)

- Teacher writes a simple chain on the board:
  bug → cause → fix → result
- Students create a short example together verbally:
  - “The login page froze → caused by an expired token → fixed by refreshing the token logic → result: load time improved.”

Warm-Up (10 mins)
Log Snapshot Guessing Game
Teacher shows short log:

```
Error 503: Service Unavailable
Timeout after 5000ms
```

Students guess:

- What might be wrong?
- What might fix it?

Teacher makes corrections on language, don’t worry about technical accuracy.

I. Presentation (30 mins)
Vocabulary Introduction (10 mins)

- Teacher writes 5 terms: *fix, patch, regression, verify, root cause*.
- Pairs match them to examples displayed on the board.
- Then students create one short spoken sentence with a term.

Reading Input: Example Fix Report (10 mins)
Teacher displays short professional sample:

Issue: Checkout page showing blank screen.
Root Cause: Null pointer in price-calculation function.
Fix Implemented: Added validation + updated error handler.
Verification: Tests passed in staging; logs show no more null pointer errors.
Result: Checkout loads correctly; no failures in last 12 hours.

Students identify:

- what the problem was
- what caused it
- what the fix was
- how they verified it

Useful Phrases for Fix Reports (10 mins)

Function	Phrases
Describing problem	“The issue occurred when…”, “Users reported that…”
Describing investigation	“We investigated and found…”, “The logs showed…”
Describing fix	“We applied a fix by…”, “We updated the…”
Showing verification	“We verified the fix in…”, “After applying the patch…”
Describing results	“As a result…”, “The system is now…”

- Project or write the above phrases on the board
- Say them out loud for students to hear correct pronunciation and emphasis
- Students work in pairs to create one fix explanation using at least two phrases.

II. Practice (40 mins)
Before/After Logs (20 mins)
Students receive a pair of logs:
Before:

```
GET /user 500 Internal Server Error
Timeout waiting for DB response
```

After:

```
GET /user 200 OK
Response time: 120ms
```

Task in pairs:

- describe the problem
- describe the root cause
- describe the fix
- describe the evidence of improvement

Teacher collects one example per group.

Write a Fix Report (20 mins)
Students receive the **Fix Report Template**:

- Problem description
- Steps to reproduce
- Root cause
- Fix implemented
- Verification steps
- Before/After data

Teacher circulates, helping students structure their writing.

Break (20 mins)

III. Production (40 mins)

Mini-Project: Issue Log + Fix Report → Presentation
Students choose one of the following options (or bring their own):
Options:

1. Slow dashboard loading
2. Intermittent login failures
3. Crash during file upload
4. Notification emails not sent
5. Search results not updating

Stage 1 — Write
Students write a full fix report using the template.
Stage 2 — Present
Students present in groups of 3–4:

1. Problem
2. Root cause
3. Fix
4. Verification
5. Result

Stage 3 — Q&A
Listeners ask clarifying questions:

- "How did you know the fix worked?"
- "What changed in the logs?"

Observer Checklist

- Clear structure (problem → root cause → fix → verification)
- Used technical language appropriately
- Answered clarifying questions clearly
- Gave evidence of improvement

IV. Wrap-Up (10 mins)
One-Sentence Summary
Students complete one sentence:
"The most important part of explaining a fix is ___."
Share 3–4 examples.

Optional Independent Practice
Write a real or fictional fix report for an issue from your own work (8–10 sentences).
Focus on:

- Root cause
- Fix
- Evidence

Teacher Notes

- Keep explanations simple; content accuracy is less important than **clear structure**.
- If students choose topics that are too complex, prompt them to narrow the scope.
- Encourage verification details — logs, metrics, tests — even if fictional.
- During Q&A, remind students to ask **clarifying, not technical** questions unless comfortable.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 25
Topic: Common Interview Questions in Tech
Lesson Duration: 3 hours (1 hr 20 – break 20 mins – 1 hr 20)
Can-Do Objectives (Aligned with CEFR Descriptors):

- Can respond to predictable interview questions clearly and confidently
- Can describe projects and achievements in structured detail
- Can explain challenges and solutions with professional tone

Materials

- Handout: "Top 10 Tech Interview Questions"
- Example answers and structure models (STAR method)
- Phrase bank for describing achievements
- Interview question cards for pair practice
- Mini-project guide and peer checklist
- Optional recording device (phone or laptop microphone)

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
achievement*	something successfully completed	"My biggest achievement was leading a refactoring project."	N
challenge*	a difficult situation or task	"A big challenge was managing legacy code."	N
deadline*	the time something must be finished	"We met every deadline during the sprint."	Y
impact*	measurable effect or result	"Our redesign had a strong impact on usability."	Y
initiative*	the ability to start things independently	"I took initiative to automate our tests."	N
adaptability*	ability to adjust to new conditions	"Adaptability is key in fast-changing tech."	N
collaboration	working well with others	"Good collaboration helped us deliver on time."	Y

framework*	a structure for organizing code or work	“We built the feature using the Django framework.”	Y
mentorship	guiding or supporting another person	“I provide mentorship to junior developers.”	Y
reflection*	serious thinking about past experiences	“Reflection helps me learn from each project.”	N

Lesson Structure (PPP)

Review (10 mins): First Impressions

1. Ask: “What’s the first thing an interviewer usually asks?” (Answer: “Tell me about yourself.”)
2. Have students share 1–2 sentences about their background with a partner.
3. Write a strong model on the board:

   “I’m a software engineer with three years of experience in backend development. I focus on building efficient and scalable APIs.”

Expected Output: Short spoken introductions with clear structure and focus.

Warm-Up (10 mins): “Speed Interview” Pairs

1. Arrange students in two lines facing each other.
2. Each pair has 1 minute to ask and answer a simple question (e.g., “What’s your strength?” “Why did you choose software engineering?”).
3. Rotate partners every minute.

Follow-Up: Ask students what felt easy or difficult.

I. Presentation (30 mins)

Reading Input: Top Tech Interview Questions (15 mins)
Text:

1. Tell me about yourself.
2. What project are you most proud of?
3. Describe a challenge you faced and how you solved it.
4. What tools or technologies do you use most often?
5. How do you handle tight deadlines or pressure?

Task: In pairs, students choose one question and underline useful keywords (*project, challenge, handle, tools*). Discuss what the interviewer wants to learn from each question.
Teacher Notes: Point out that interview questions often test communication, not only technical skill.

Model Structure: STAR Method (15 mins)
Situation – Describe the context. \n**T**ask – Explain your role or goal. \n**A**ction – Describe what you did. \n**R**esult – Share the outcome.
Example:

> "Last year, our login page crashed frequently (**Situation**). I was responsible for fixing it (**Task**). I implemented a new API handler (**Action**), which reduced failures by 50% (**Result**)."

Practice: Have students identify each section in the example and label it S, T, A, R.

II. Practice (40 mins)

Phrase Matching (15 mins)

Function	Example Phrases
Starting Answer	"Sure, let me start with…", "I'd be happy to share an example…"
Describing Work	"My main responsibility was…", "I worked closely with…"
Explaining Action	"To solve this, I decided to…", "I implemented…"
Describing Results	"As a result,…", "This improved performance by …"
Reflecting	"From this experience, I learned that …"

Task: Students match each function to a sentence from the model answer, then create their own mini answer for one question.

Interview Cards (25 mins)

1. Give each pair one card at a time. Student A asks, Student B answers using the STAR framework.
2. Switch roles and rotate cards.

Sample Cards:

- "Tell me about a time you improved a process."
- "Describe a technical problem you solved recently."
- "What's a challenge you faced working in a team?"
- "How do you keep your skills up to date?"
- "Give an example of a project that didn't go as planned."

Teacher Tip: Model one short answer before starting so students see timing and structure.

III. Digital Tool (45 mins)

- To provide students the opportunity to use the digital tool in class with teacher support.

- The teacher can demonstrate activities with the whole class and/or support students as they work individually.

IV. Wrap-Up (15 mins)

Reflection Prompts:

- Which questions felt easy? Which were hard?
- How did the STAR method help you?
- Which phrases will you reuse in a real interview?

Quick Recap Game:

Teacher says a key word (*challenge, result, impact*). Students give a 1-sentence example from their own experience.

"Challenge – We had to finish the feature in 2 days."

"Result – We delivered on time and the client was happy."

Optional Independent Practice

Instructions for Students: Write a short (150-200 word) answer to one of the following questions:

- Tell me about yourself.
- Describe a project you're proud of.
- What's a challenge you've overcome in your career?

Model Answer:

"I'm a backend developer with four years of experience in Python and Django. Last year, I led a project to improve our API speed by 30%. It was challenging because we had tight deadlines, but I analyzed bottlenecks and refactored the code. As a result, performance and client satisfaction improved. I learned the value of testing and clear communication."

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 26
Topic: Talking About Strengths and Weaknesses
Lesson Duration: 3 hours (1hr20 – break 20 mins – 1hr20)
Can-Do Objectives (Aligned with CEFR descriptors):
Can describe strengths with supporting examples
Can describe weaknesses with strategies to improve
Can speak confidently about professional growth
Materials
Handout: Strengths & Growth worksheet (3 strengths / 2 growth areas / evidence column)
Phrase-bank handout for strengths/weakness language
Interview prompt cards (strengths & weaknesses)
Observer checklist (for peer feedback)
Jamboard/Padlet or a whiteboard for shared reflections
Optional: phone or laptop for short recordings (Flipgrid/Loom)

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
strength*	a skill or ability someone does well	"One of my strengths is debugging intermittent errors."	N
weakness*	an area that needs improvement	"A weakness of mine is public speaking in large meetings."	N
initiative*	taking action without being told	"She showed initiative by automating the tests."	Y
resilience*	ability to recover and keep working after setbacks	"His resilience helped the team meet the deadline."	N
adaptability*	ability to adjust to new tools or circumstances	"Adaptability helped her learn the new framework quickly."	Y
attention to detail	carefulness that prevents mistakes	"Attention to detail reduced production bugs."	Y
accountability	taking responsibility for results	"He showed accountability by owning the bug fix."	N

overcommit	to take on too many tasks	“I tend to overcommit when the sprint is busy.”	N
growth mindset	belief that abilities can improve with effort	“A growth mindset helped him learn testing tools.”	N
constructive*	helpful and solution-focused feedback	“She gave constructive feedback during the retro.”	N

Lesson Structure (PPP)

Review (10 mins)
Quick Recall — STAR Refresher

- What’s the best way to explain a strength or achievement?
- Teacher elicits STAR (Situation, Task, Action, Result) and writes a one-line model on the board.
- Students quickly turn the model into one sentence describing an achievement (spoken to a partner).
- Aim: reconnect language pattern used in interviews and past lessons.

Teacher steps: 1) Show one strong example; 2) students tell partner their own 1-sentence version; 3) collect 1–2 examples aloud.

Warm-Up (10 mins)
Two Truths + One Growth Area (Fast Pair)

- Students quickly tell a partner: two real strengths and one growth area. Partner guesses the growth area and asks one follow-up question (e.g., “How did you use that strength?”). Rotate once.

Teacher steps: Model one example, monitor pairs, note good language to share in debrief.

I. Presentation (30 mins)

Input: Model Answers + Language Frames

1. Project or hand out two model texts (one strength, one weakness → growth).
    - Strength model (STAR condensed): “I led a migration (Situation). I was responsible for refactoring the auth module (Task). I split the work, wrote tests and monitored metrics (Action). As a result, errors dropped 40% and deployments were smoother (Result).”
    - Weakness → growth model: “I used to avoid asking for help (S/T). I realized this slowed us down, so I now schedule short pairing sessions (A). This helped me improve speed and team knowledge sharing (R).”
2. Highlight and explain useful frames on board:
    - “One of my strengths is…”
      “I’m strong at…, which helped… ”

    - "I used to…, but now I…"
    - "I'm improving by…"
    - "As a result, … (impact metric or benefit)."
3. Quick choral drill of 5 frames for pronunciation and rhythm.

Expected output: Students can identify S/T/A/R parts and copy 4 useful frames.

II. Practice (30 mins)

Strength Evidence Grid

- Students complete the handout: list 3 strengths, for each give a short example (S/T/A/R) and one evidence/impact (metric, user benefit, team outcome). Teacher models one example.

Flip the Frame — Weakness → Growth

- Pairs: Student A states a real weakness aloud. Student B helps reframe it into a growth statement and suggests a concrete action. Swap roles.

Teacher steps: circulate, supply language if stuck, write 3 strong reframes on board.
Expected output: Each student produces 3 STAR strength mini-statements and 2 reframed growth statements with actions.

Break (20 min)

III. Production (50 mins)

Mock Interview Rotation + Peer Feedback (40 mins)
Structure: groups of three (Interviewer / Candidate / Observer). Each round = 10 minutes interview + 6 minutes feedback = 16 minutes; do three rounds ≈ 48–50 minutes.
Round details:

- Interviewer asks two prompt cards (one about strength, one about weakness/growth). Candidate answers using STAR + growth framing. Observer completes the checklist and gives short feedback (warmth + 1 suggestion).

Sample prompt cards:

- "What is a professional strength that helps you in engineering? Give a short example."
- "Tell me about a weakness you are working to improve and what you do about it."
- "Describe a time you failed or missed a deadline. What did you learn?"
- "How do you make sure you don't overcommit during a sprint?"

Observer checklist (short):

- S/T/A/R present? (Y/N)
- Clear example + impact (Y/N)
- Tone: confident but humble (Y/N)
- One concrete suggestion given

1) Model one interview
2) start rotation timer
3) monitor and note strong answers to share.
Expected output: Each student completes 3 interview turns (2 as candidate) and receives structured peer feedback.

Peer Feedback & Micro-Teachable (20 mins — included in Production block)
After rotations, each student receives 2 peer suggestions and chooses one improvement to practise. Volunteers deliver a 30–45 second improved version to the class. Teacher highlights language that improved.

IV. Wrap-Up (10 mins)
Activity: Quote Yourself + Next Steps

- Each student writes one short sentence for the class Jamboard: “My interview strength is ___; my next step is ___.”
- Teacher reads 4–6 aloud and gives brief positive feedback. Quick reminder of optional independent practice.

Optional Independent Practice

- Record a 60–90 second video (Flipgrid/Loom) answering: “Describe one strength and one growth area (use STAR).”
- Upload and optionally exchange with one peer for feedback.
- Teachers can grade for fluency and use of frames.

Notes for Instructor

- Keep the tone supportive: encourage honest but strategic framing of weaknesses.
- Model reframing; give sentence frames often.
- If students struggle with tech examples, allow non-technical workplace examples but ask them to relate to team impact.
- Timekeeping is important during the production rotation — use a visible timer and give 30s warning.
- Encourage observers to provide one positive comment and one specific suggestion (not general).

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 27
Topic: Answering Scenario-Based Questions
Lesson Duration: 3 hours (1hr20 – break 20 mins – 1hr20)
Can-Do Objectives (Aligned with CEFR descriptors):
Can structure answers using Situation–Task–Action–Result
Can explain reasoning in problem-solving scenarios
Can adapt examples to highlight relevant skills
Materials
Scenario question cards (easy → challenging)
STAR method reminder sheet
Decision-making phrase bank
Problem → Options → Decision → Justification worksheet
Observer checklist
Whiteboard or slides

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
hypothetical	imagined situation	"In a hypothetical outage, I would restart the service."	N
scenario	a possible situation or case	"In this scenario, the deployment fails halfway."	N
constraint	a limitation affecting choices	"We worked under strict time constraints."	N
trade-off*	balance between two competing factors	"There is a trade-off between speed and stability."	Y
priority*	the most important task	"My priority would be restoring service."	Y
escalate*	to raise an issue to someone senior	"I escalated the issue to the DevOps lead."	Y
allocate	to assign resources appropriately	"We allocated two engineers to debugging."	Y
mitigation	action that reduces the impact of a problem	"Caching was used as a mitigation strategy."	Y

root cause*	the deeper reason for the problem	“The root cause was missing configuration.”	Y
outcome*	the final result	“The outcome was faster load time.”	N

Review (10 mins)
Activity: STAR Quick Recall
Students quickly tell a partner an achievement using the STAR structure.
Goal: reactivate structure before applying it to hypothetical situations.

Warm-Up (10 mins)
Activity: Lightning Round – “What Would You Do If…?”
Teacher gives easy everyday workplace hypotheticals:

- “What would you do if your Wi-Fi stopped working before a meeting?”
- “What would you do if a teammate disagreed with you?”

Students answer in **one quick sentence** (not STAR yet).
Purpose: shift students into hypothetical problem-solving mode.

I. Presentation (30 mins)

Vocabulary Introduction (10 mins)
Match & Predict (Pairs)

1. Teacher writes five vocabulary terms on the board: *scenario, constraint, escalate*, *trade-off*, *mitigation*.
2. Students guess meanings and match them to simple examples.
3. Teacher confirms definitions and asks students to create one short spoken example.

Expected Output:
“A constraint is a limit, like time or resources.”
“A trade-off is when you choose speed over memory.”

Reading Input: Model Scenario Answer (10 mins)
Scenario Question:
“Imagine the release of a new app is at risk because of a last-minute bug. What would you do?”
Model Answer:

- **Situation:** “This happened last quarter during a Friday release.”
- **Task:** “My task was to assess the bug’s impact quickly.”
- **Action:** “I reproduced the issue, escalated* it to QA, and worked with a teammate to patch it.”
- **Result:** “We fixed it in two hours and the release went out on time.”
  Students label each part S–T–A–R.

Useful Phrases for Scenario Answers (10 mins)

Function	Example Phrases
Explaining reasoning	“I chose this approach because…”, “The key factor was…”
Exploring options	“One option is…, another option is…”
Making decisions	“I would decide to…”, “My first step would be…”
Showing priorities	“My main priority would be…”
Describing outcomes	“This would help us…”, “This reduces the risk of…”

Pairs create **one short answer** using two phrases.

II. Practice (40 mins)

Scenario Sort — Easy, Medium, Hard (20 mins)
Groups receive scenario cards and sort them by difficulty.
Examples:

- *Easy:* “A teammate sends you an unclear Slack message.”
- *Medium:* “A feature works locally but fails in staging.”
- *Hard:* “Traffic spikes cause the API to timeout during peak usage.”

Groups choose one from each level and draft a **bullet STAR outline**.

Reasoning Pathways (20 mins)
Students complete a worksheet:
Problem → Options → Decision → Justification
Teacher models:

- Problem: “Production login fails”
- Options: restart service / roll back / debug live
- Decision: roll back
- Justification: fastest + lowest risk

Pairs complete one full reasoning pathway based on a scenario they chose.

Break (20 mins)

III. Production (40 mins)

Mock Scenario Interview Rotation
Groups of three rotate:

1. **Interviewer** – chooses question card
2. **Candidate** – answers using STAR + reasoning
3. **Observer** – marks checklist

Each round: **6 minutes interview + 6 minutes feedback**
Scenario Card Examples:

- “You notice a performance issue during a client demo. What do you do?”
- “A teammate insists on a solution you believe won’t scale. How do you handle it?”
- “The production server goes down an hour before launch. What is your first step?”
- “What would you do if a stakeholder demands a feature the team can’t finish in time?”

Observer Checklist

- Clear STAR structure (Y/N)
- Reasoning shown (Y/N)
- Options considered (Y/N)
- Professional tone (Y/N)
- Logical or relevant approach (Y/N)

Teacher circulates, supporting non-technical instructors with guidance on **clarity**, not technical accuracy.

IV. Wrap-Up (10 mins)

Best Line Rewrite

- Students choose one sentence from their answer and rewrite it to make it *clearer, more structured, or more confident*.

Example rewrite:

Original: “I would try to fix it.”

Improved: “My first step would be to reproduce the issue to understand its impact.”

Students share aloud; teacher highlights strong language.

Optional Independent Practice

- Students write one full STAR answer (8–10 sentences) to a difficult scenario card and bring it next class for feedback.

Teacher Notes

- Simplicity over correctness: focus on clear English and strong structure, not technical depth.
- Encourage use of *multiple* options (“One option is…, another is…”) before choosing one.
- When students give unrealistic technical answers, redirect gently:
  “How can you show teamwork / decision-making / communication more clearly here?”
- Keep pacing tight during rotations — use a visible timer.
- Praise good structure: clear reasoning, organized STAR, confident tone.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 28
Topic: Reflecting on Career Growth
Lesson Duration: 3 hours (1hr20 – break 20 mins – 1hr20)**
Can-Do Objectives (Aligned with CEFR descriptors):

- Can describe progress in skills and roles
- Can set short-term and long-term goals
- Can describe what they want to learn next

Materials

- Career Timeline worksheet
- Phrase bank for reflection and goal-setting
- Growth reflection cards
- Goal-setting template
- Observer checklist for peer conversations
- Whiteboard or digital board

Vocabulary

Term	Definition	Example Sentence	Profession-Specific (Y/N)
milestone	an important point in progress	"Completing my first major feature was a milestone for me."	N
progress*	improvement over time	"I've made progress in writing clean code."	N
long-term	happening over a long period	"My long-term goal is to lead a small development team."	N
short-term	happening soon or in the near future	"My short-term goal is to improve my cloud skills."	N
aspiration	something you want to achieve	"My aspiration is to specialize in backend security."	N
competence	the ability to do something well	"My competence in debugging has grown a lot."	N
initiative*	action taken without being asked	"I showed initiative by proposing a documentation update."	Y
evolve	to change or grow gradually	"My role evolved from bug fixing to designing small features."	N

roadmap	a plan for future development	“We created a roadmap for learning Kubernetes.”	Y
mentor*	a person who helps guide learning	“My mentor helped me understand architectural decisions.”	Y

Review (10 mins)
“Progress Check-In” (Pair Quick Talk)
Students tell partners one thing they’ve improved in the last year (work or personal).
Partners ask:

- “How did you improve it?”
- “How do you know you improved?”

Teacher collects 2–3 examples to highlight the theme of growth.

Warm-Up (10 mins)
Past → Present → Future Line-Up
Teacher writes three prompts on the board:

- “One thing I used to struggle with…”
- “One thing I’ve improved recently…”
- “One thing I want to learn next…”

Students stand and physically line up left→right representing past → present → future.
Students share answers with the person next to them.
Purpose: get students thinking in timelines.

I. Presentation (30 mins)

Vocabulary Introduction (10 mins)
Term Sorting – Growth or Planning?
Teacher writes eight terms on the board: milestone, progress*, long-term, short-term, aspiration, initiative*, evolve, roadmap.
Pairs sort them into two groups:

- **Growth Words**
- **Planning Words**

Teacher checks as a class and gives short definitions.
Students say one sentence aloud using a word from each category.

Input Model: Career Growth Reflection (10 mins)

- Teacher projects short text:
  “When I started, I mainly fixed bugs. Over the past year, I’ve grown into writing new features and mentoring interns. A major milestone was rewriting our login flow. My short-term goal is to improve my system design skills, and long-term I want to become a team lead.”
- Students highlight:

- Growth
- Milestones
- short-term goals
- long-term goals

Useful Phrases (10 mins)

Function	Example Phrases
Describing growth	"I used to…, but now I…", "I've become more confident in…"
Naming milestones	"A key milestone was…", "One important step in my career was…"
Setting short-term goals	"In the next few months, I want to…", "I'm focusing on improving…"
Setting long-term goals	"Eventually, I'd like to…", "My long-term aspiration is to…"
Describing learning plans	"I plan to learn…, starting with…", "I want to develop competence in…"

- Project the above table on the board or send to the class group chat
- Students work in pairs and choose one phrase from each row and build a short spoken paragraph.
- Ask students to share their paragraph with the class

II. Practice (40 mins)

Career Timeline Worksheet (20 mins)

Students fill in a timeline with four sections:

- Past role/skill level
- Progress & milestones
- Current strengths
- Future goals (short-term & long-term)

Teacher models a simple example.

Pairs share and ask each other:

- "Which milestone are you most proud of?"
- "Which goal feels most important right now?"

Growth Reflection Cards (20 mins)

Students draw a card with a prompt such as:

- "Describe a milestone you reached in the last two years."
- "How has your responsibility evolved?"
- "Describe a learning moment you didn't expect."
- "What skill do you want to be known for in three years?"

Students answer using **growth phrases** and **goal-setting language**.
Teacher circulates, notes strong examples, and supports hesitant speakers.

Break (20 mins)

III. Production (40 mins)

Mini-Project: Career Development Conversation

- Groups of 3 take roles:
    1. **Employee** – shares career growth and goals
    2. **Manager** – asks guiding questions (provided by teacher)
    3. **Observer** – uses checklist to evaluate clarity, confidence, and detail
- Students converse as their roles
- Teacher moves around the room listening for correct grammar and vocabulary use

Manager Question Prompts

- "Which skills have grown the most this year?"
- "What roles or tasks would you like to take on next?"
- "What challenges are you preparing for?"
- "What support do you need from your team?"

Observer Checklist

- Used at least one growth phrase
- Mentioned past → present → future
- Provided at least one milestone
- Named one short-term goal
- Named one long-term goal
- Spoke clearly and confidently

Rounds:
10 minutes conversation + 6 minutes feedback × 3 rotations.

IV. Wrap-Up (10 mins)

One Sentence Future Vision
Students write one sentence beginning with:

- "In one year, I want to be able to…"
- OR "I'm motivated to grow because…"

Students share 3–4 examples aloud.

Optional Independent Practice

- Write a **150–200 word reflection** including:
    - one milestone
    - one area of progress
    - one short-term goal
    - one long-term goal
    - what you want to learn next

Teacher may give written feedback using the observer checklist.

Teacher Notes

- Prioritize clarity over technical detail.
- Students may choose real or imaginary growth examples — authenticity is good, but fluency and structure matter more.
- Encourage specific, measurable goals (avoid “get better at coding”; aim for “learn Docker basics” etc.).
- If students struggle, help them anchor milestones to projects, responsibilities, or skills learned.
- Ensure all three stages (past → present → future) appear in their spoken and written work.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 29
Topic: Selecting a Capstone Topic
Lesson Duration: 3 hours (1hr20 – break 20 mins – 1hr20)
Can-Do Objectives (Aligned with CEFR descriptors):

- Can choose a relevant, clear topic with sufficient depth
- Can outline their talk using logical sections
- Can adapt their message for a technical or mixed audience

Materials

- Topic brainstorm cards
- Audience profiles (tech / mixed / non-technical)
- Phrase bank for outlining and framing presentations
- Capstone planning worksheet
- Optional: Jamboard / Miro for collaborative outlining

Vocabulary
(*Asterisk = previously introduced word*)

Term	Definition	Example Sentence	Profession-Specific (Y/N)
scope	the range of content included	"We narrowed the scope to focus on the API layer."	Y
relevance*	how useful or connected something is	"This topic has direct relevance to the onboarding process."	N
audience*	the group receiving the message	"The audience is a mix of PMs and junior engineers."	N
overview	a summary of the main points	"I'll begin with an overview of the system."	N
section	a divided part of content	"The first section explains the workflow."	N
rationale	the reasoning behind a choice	"Our rationale was to reduce load time."	N
clarify*	to make something easier to understand	"I clarified the diagram to help the PM understand."	N
emphasize	to stress or highlight something	"I want to emphasize scalability improvements."	N

objective	the purpose or goal	“The objective of my presentation is to explain the data flow.”	N
summary	a brief final statement	“I’ll end with a short summary and key insights.”	N

Review (10 mins)

- **Activity: “What Have We Learned?” Pair Recap**
  Each student tells a partner two lessons/topics they feel most confident presenting (e.g., system diagrams, optimization, collaboration, bugs, workflows).
- Partners respond with:
  - “Why do you think that’s a strong topic?”
  - “Who would benefit from hearing it?”
- Teacher takes 2–3 examples to lead into the idea of selecting a strong capstone topic.

Warm-Up (10 mins)
Topic Hits & Misses

- Teacher writes two lists on the board:
  - **Strong Capstone Topics**
    - Explaining a system
    - A feature or workflow
    - An optimization or improvement
    - A debugging process
    - A team communication strategy
  - **Weak Capstone Topics**
    - “English learning journey”
    - “Why I like programming”
    - “My favorite tool with no example”
    - “Random personal story”
- Students discuss in pairs: *What makes a topic strong or weak?*.
- Teacher confirms: strong topics are specific, explainable, audience-relevant, and allow structure.

I. Presentation (30 mins)

Vocabulary Introduction (10 mins)
Categorize the Terms — Topic Selection or Presentation Structure?

- Teacher writes the vocabulary list on the board.
- Pairs sort terms into two categories:
  1. Topic Selection: scope, relevance, rationale, audience*
  2. Presentation Structure: overview, section*, clarify*, emphasize, objective*, summary
- Students share one sentence using one term from each category.

Input Model: Sample Capstone Topics (10 mins)
Teacher shows three example capstone options:

1. **System Explanation:** "How our login system works (architecture + data flow)"
2. **Workflow Walkthrough:** "How we handle bug triage in our sprint process"
3. **Improvement Case Study:** "Reducing API latency by 40%: problem → change → outcome"

Discussion questions:

- "Who is the audience for each topic?"
- "Which has the clearest scope?"
- "Which one feels most interesting?"

Useful Phrases for Topic Framing (10 mins)

Function	Example Phrases
Introducing a topic	"The focus of my presentation will be…"
Describing scope	"I'll limit the scope to…", "I'll focus mainly on…"
Explaining relevance	"This is relevant because…", "This helps the team understand…"
Outlining structure	"My presentation has three main sections…", "First…, then…, finally…"
Adapting to audience	"For a mixed audience, I'll emphasize…", "For engineers, I'll dig deeper into…"

- Project the above table on the board or send to class group chat
- Pairs choose one topic idea and say two sentences using these phrases.

II. Practice (40 mins)
Topic Brainstorm Carousel (20 mins)

- Groups move around stations labeled:
  - System / Architecture
  - Workflow or Process
  - Optimization / Improvement
  - Debugging Case
  - Team Communication Strategy
  - Tools or Collaboration Platform
- At each station, students brainstorm possible capstone topics.
- Each group writes 1–2 ideas before rotating.
- Teacher gathers the strongest topics on the board.

Audience Adaptation (20 mins)
Students receive audience profiles:

- **T**echnical audience: engineers, architects
- Mixed audience: PMs, designers, engineers
- Non-technical audience: clients, leadership

Pairs choose one topic and adjust how they'd present it to each audience:

- What to emphasize
- What to simplify
- What to omit
- What visuals to include

Teacher circulates and prompts adjustments.

Break (20 mins)

III. Production (40 mins)

Mini-Project: Capstone Topic Selection + Outline

Students choose **1 final capstone topic**.

Using the Capstone Planning Worksheet, they must create:

1. Topic Statement
2. Audience & Rationale
3. Scope (what's included / what's not)
4. 3–4 Section Outline
5. One sentence for each section
6. One adaptation choice (what they'll adjust for their chosen audience)

Pairs review each other's outlines using a checklist:

Peer Checklist

- Topic is specific and clear
- Topic has enough depth for 5–7 minutes
- Outline has logical sections
- Audience is clearly identified
- Scope is realistic
- Adjustments for audience make sense

Teacher collects 1–2 outlines to read aloud as strong models.

IV. Wrap-Up (10 mins)

One-Sentence Pitch

- Students stand in a circle. Each gives a one-sentence pitch for their chosen capstone topic.

Teacher gives light feedback on clarity and scope.

Optional Independent Practice

Students refine their outline into a **1-paragraph project proposal** including:

- topic
- audience
- relevance

- outline summary
- why they chose it

Teacher may provide written comments before Lesson 29.

Teacher Notes

- Keep students focused on specific, explainable, and structured topics, not broad or personal themes.
- Remind them that the capstone must be something they can explain clearly with the tools from previous units (workflow language, diagrams, comparisons, reasoning, etc.).
- If students choose topics that are too technical, help them narrow the scope to one workflow or feature.
- Encourage early thinking about visuals, examples, and reasoning, because these will matter in Lessons 29–30.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 30
Topic: Building the Presentation
Lesson Duration: 3 hours (1hr20 – break 20 mins – 1hr20)**
Can-Do Objectives (Aligned with CEFR descriptors):

- Can support presentation points with clear visuals and examples
- Can use transition language to maintain flow
- Can keep the presentation organized, clear, and within time limits

Materials

- Sample slide deck (simple, clean design)
- Visual design guidelines sheet
- Transition phrase bank
- Slide planning worksheet
- Timer (for rehearsal)
- Peer feedback checklist

Optional digital tools: Google Slides, Canva, Miro, Lucidchart

Vocabulary
(*asterisk = previously introduced word*)

Term	Definition	Example Sentence	Profession-Specific (Y/N)
transition	a phrase that links sections smoothly	"My next point is…, which leads us to…"	N
visual hierarchy	organization of elements by importance	"Use visual hierarchy to highlight key messages."	N
readability*	how easy text is to read	"Increase contrast for better readability."	N
diagram*	a simple visual representation	"This diagram shows the data flow."	Y
chart	a visual representation of data	"The chart highlights performance changes."	N
highlight*	emphasize or draw attention to	"I'll highlight the optimization results."	N
pace	the speed of speaking and presenting	"Keep a steady pace to maintain clarity."	N

timing	managing how long each section takes	“Your timing was perfect—6 minutes total.”	N
emphasis*	special importance placed on an idea	“The emphasis here is on user impact.”	N
summary*	a final restatement of main ideas	“In my summary, I returned to the key outcome.”	N

Review (10 mins)
“Show Me the Structure”
Teacher displays a short presentation text (5–6 sentences).
Students identify:

- introduction
- section 1
- transition
- section 2
- conclusion

Pairs discuss: *Which part is the strongest? Why?*

Warm-Up (10 mins)
Speed Critique
Teacher presents **two different ways**:

- one clear, easily understandable, simple
- one speaking too fast, looking down, confusing wording

Students vote:

- Which way is better?
- What rules does it follow or break?

Teacher writes class-generated rules on the board.

I. Presentation (30 mins)
Vocabulary Introduction (10 mins)

- Teacher writes these terms on the board:
  - *transition, highlight, diagram, chart, readability, pace, emphasis, summary*
- Pairs sort them into two groups:
  - **Group A – Visuals:** diagram, chart, readability, emphasis
  - **Group B – Delivery:** transition, pace, summary, highlight
- Students say one example sentence for any two terms.

Useful Phrases (10 mins)

Function	Example Phrases
Starting a section	“Let’s begin with…”, “I’d like to introduce…”

Moving on	“Now let’s move on to…”, “The next point is…”
Explaining visuals	“This diagram shows…”, “As the chart highlights…”
Emphasizing	“What’s important here is…”, “A key takeaway is…”
Concluding	“To summarize…”, “In conclusion…”

- Pairs choose one section transition and one visual explanation phrase to combine into a mini practice sentence.
- Pairs practice with each other
- Pairs turn to another pair next to them and present their practice sentences.

II. Practice (40 mins)

Slide Planning Worksheet (20 mins)

Students complete a three-part worksheet:

1. **Slide 1 – Introduction**
    - Title
    - One sentence objective
2. **Slide 2 – Main Section**
    - Diagram or bullet outline
    - Example or data point
3. **Slide 3 – Conclusion**
    - Summary point
    - Final takeaway

Teacher circulates, ensuring slides are simple and readable.

Transitions + Timing Rehearsal (20 mins)

Pairs practice:

- reading their introduction aloud
- using at least **two transition phrases**
- adding one explanation of a visual

Each student rehearses for **45–60 seconds**.

Partner feedback questions:

- “Was the pace too fast, slow, or just right?”
- “Was the transition smooth?”
- “Did the visual help?”

Teacher models correct pacing if needed.

Break (20 mins)

III. Digital Tool (45 mins)

- To provide students the opportunity to use the digital tool in class with teacher support.

- The teacher can demonstrate activities with the whole class and/or support students as they work individually.

IV. Wrap-Up (10 mins)
One Slide, One Improvement

- Students choose one slide they created and write **one improvement** they will make before next class.
- Share 3–4 examples aloud.

Optional Independent Practice
Students refine their full deck (5–7 slides) including:

- intro slide
- 3–4 content slides
- conclusion slide
- speaker notes

Encourage them to use diagrams or visuals wherever possible.

Teacher Notes

- Remind students that visuals should support, not replace, spoken explanation.
- If students over-explain, prompt them to shorten and highlight key points.
- Encourage consistent pacing — avoid rushing slides.
- Focus feedback on clarity + structure, not technical accuracy.
- Lesson 31 will be full presentation rehearsal + feedback, so completeness matters today.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 31
Topic: Rehearsal + Peer Feedback
Lesson Duration: 3 hours (1hr20 – break 20 mins – 1hr20)
Can-Do Objectives (Aligned with CEFR descriptors):

- Can present confidently with minimal notes
- Can give and receive useful, specific peer feedback
- Can revise a presentation to improve clarity, structure, tone, or flow

Materials

- Peer feedback forms
- Timer (phone or online tool)
- Transition phrase bank (reuse from Lesson 30)
- Delivery checklist
- Optional: phones for video recording
- Optional: Google Slides, Canva, Miro

Vocabulary
(*asterisk = previously introduced word*)

Term	Definition	Example Sentence	Profession-Specific (Y/N)
delivery*	the way a presentation is spoken	"Your delivery was clear and steady."	N
flow*	how smoothly ideas connect	"Your flow improved after adding transitions."	N
clarity*	how easy something is to understand	"Your clarity increased when you simplified the diagram."	N
tone*	the speaker's attitude or style	"Your tone was appropriate for a mixed audience."	N
confidence*	speaking comfortably and clearly	"Your confidence grew on the second try."	N
gesture	movement used to support communication	"Use gestures to highlight keywords."	N
eye contact	looking at the audience to engage them	"Strong eye contact builds trust."	N
pacing	the speed at which someone speaks	"Your pacing was consistent throughout."	N

posture	body position that communicates confidence	“Stand with open posture while presenting.”	N
revision*	improving something by changing parts of it	“After feedback, I made revisions to my introduction.”	N

Review (10 mins)
Activity: “Delivery Do’s and Don’ts”
Teacher writes two columns on the board:
Do and **Don’t**
Students brainstorm examples:

- Do: speak clearly, slow down, make eye contact
- Don’t: read entire slides, rush, speak in monotone

Teacher adds 1–2 important missing points.

Warm-Up (10 mins)
30-Second Elevator Pitch Swap
Students pair up and take turns giving a **30-second version** of their capstone intro.
Partners ask:

- “What part was strongest?”
- “What part needs more clarity?”

Then students switch partners once.
Purpose: warm up voices + review intro sections.

I. Presentation (30 mins)
Vocabulary Introduction (10 mins)
Teacher says or models a short mini-presentation (20–25 seconds).
“Good morning everyone.
Today I want to explain how our team plans weekly meetings.
First, we review last week’s tasks.
Then, we discuss priorities for the next few days.
Finally, we decide who is responsible for each task.
The goal is to keep everyone aligned and avoid confusion.
Thank you.”
Students identify vocabulary terms:

- tone
- pacing
- gesture
- clarity
- confidence
- posture

Pairs create one improvement suggestion using a vocabulary term:
“To improve pacing, you could pause between sections.”

Input: Model Peer Feedback Exchange (10 mins)

Teacher shows a written example:

Speaker:

"I focused on explaining the system flow, but I think my transitions were weak."

Peer Feedback:

"Your explanation was clear, but your pacing sped up at the end. Try pausing before the diagram."

Class identifies what made the feedback "B2-quality":

- specific
- balanced
- actionable

Useful Phrases for Giving Feedback (10 mins)

Purpose	Example Phrases
Noticing strengths	"One thing you did well was…"
Suggesting improvements	"It would be clearer if you…"
Balancing critique	"Your explanation was strong, but the pacing could be slower."
Asking for clarification	"Can you repeat how the system handles errors?"
Checking understanding	"Do you feel this part is clear to a mixed audience?"

- Send table to class group chat or project on the board
- Pairs practice giving **one positive + one suggestion** using the phrases.

II. Practice (40 mins)

Partial Rehearsal — Introduction + First Section (20 mins)

Students work in pairs and present to each other:

- Introduction
- First main section
- One visual explanation
- One transition

Time: **1–2 minutes each**

Partners use a short checklist:

- Was the objective clear?
- Was the section well structured?
- Was the visual explained?
- Was the pacing appropriate?

Swap roles.

Focused Feedback & Revision (20 mins)
Students choose **one area** to revise:

- transitions
- clarity of explanation
- timing
- slide layout
- pacing / tone

They apply one revision, then deliver the section again in pairs.
Partners must give **specific feedback using phrase bank**.
Teacher circulates and prompts deeper revisions.

Break (20 mins)

III. Production (40 mins)

Mini-Project: 3–4 Minute Rehearsal (Group Practice)
Students rehearse a half-length version of their presentation:

- Introduction
- Two main sections
- One visual
- Conclusion sentence

Total: 3–4 minutes per student
Small groups of 3–4 rotate roles:

- Presenter
- Feedback-giver
- Timekeeper
- (Optional) Recorder with phone

Group Feedback Checklist

- The speaker used transitions
- The explanation was clear to the intended audience
- Visual(s) supported understanding
- Pacing and tone were effective
- Delivery felt confident
- Flow was logical

Students revise notes based on peer comments.

IV. Wrap-Up (10 mins)

“What I Will Improve Before the Final Presentation”
Students write one delivery goal + one content goal.
Examples:

- “Pause longer between sections.”
- “Make my diagram simpler.”

2–3 volunteers share.

Optional Independent Practice
Students rehearse the full 5–7 minute presentation using a timer at home.
Optional: record themselves and note:

- pacing
- transitions
- confidence
- clarity
- slide alignment

Teacher Notes

- Focus on **delivery + clarity**, not technical correctness.
- Encourage feedback that is *specific, actionable, and kind*.
- Guide students to avoid reading from notes; aim for confident, organized speaking.
- Time each rehearsal carefully to teach pacing.
- Remind students that Lesson 32 will include final presentations + instructor feedback.

Course Title: English for Software Professionals (CEFR B2)
CEFR Level: B2
Lesson Number: Lesson 32
Topic: Final Capstone Presentations + Instructor Feedback
Lesson Duration: 3 hours (1hr20 – break 20 mins – 1hr20)**
Can-Do Objectives (Aligned with CEFR descriptors):

- Can deliver a clear, confident, well-structured presentation
- Can use visuals, transitions, and examples effectively
- Can respond to follow-up questions professionally
- Can reflect on strengths and areas for future growth

Materials

- Presentation rubric (teacher + peer versions)
- Timing sheet (5–7 minutes per student)
- Certificate of Completion
- Optional: phone or laptop for recording
- Optional: projector or screen for slides

Review (10 mins)
Final Prep Reflection
Students reflect silently for 1 minute, then pair-share:

- What is your strongest section?
- What one thing did you improve since Lesson 30/31?
- What is one worry you still have?

Teacher reassures students and clarifies expectations.

Warm-Up (10 mins)
Confidence Boost Circle
In a standing circle, each student completes:
"One thing I am proud of in my presentation is…"
"One thing I want the audience to notice is…"
This sets positive tone and reduces anxiety.

I. Presentation (30 mins)

Input: What a Strong Final Presentation Looks Like (10 mins)
Teacher briefly models a 30–40 second mini-presentation, demonstrating:

- clear structure
- transitions
- visual explanation
- confident tone
- conclusion sentence

Students identify what worked.

Review of Expectations (10 mins)

Teacher presents the **Final Presentation Checklist:**

- 5–7 minutes
- intro → sections → visual → conclusion
- transitions
- slide clarity
- Q&A professionalism
- audience-focused explanation
- steady pacing

Students ask final questions before beginning.

II. Practice (40 mins)

(Last chance to warm up before the real presentations)
Final Run – The First 30 Seconds (20 mins)
Pairs rehearse ONLY their opening section:

- introduction
- objective
- transition into first section

Partners give micro-feedback using a mini-checklist:

- Was the objective clear?
- Did the introduction feel confident?
- Was the transition smooth?

III. Production (60 mins)

Final Capstone Presentations
Each student presents **5–7 minutes**:

1. Introduction
2. Main sections
3. Diagram/visual explanation
4. Conclusion + takeaway
5. Short Q&A (1–2 questions per student)

Presentation Rubric (Teacher Use)
Scored holistically on:

- **Clarity** (structure & explanation)
- **Delivery** (tone, pacing, confidence)
- **Visuals** (support understanding)
- **Transitions** (smooth flow)
- **Audience awareness**
- **Q&A responses**

Peer Feedback (Short & Supportive):
Each peer writes **two** comments:

- “One thing you did well was…”

- "One thing to consider for next time is…"

Teacher reminds students to be positive, specific, and constructive.

IV. Wrap-Up (10 mins)

Activity: End-of-Course Reflection

- Students complete:
  "One skill I improved the most in this course was…"
  "One communication goal I have for the next 3 months is…"
- Teacher acknowledges student progress and celebrates the completion of the B2 course.

Teacher Notes

- Keep atmosphere supportive — avoid overly detailed technical critique.
- Prioritize confidence, clarity, structure, and communication strategy.
- Time presentations strictly (5–7 minutes).
- Encourage applause and positive reinforcement.
- End with congratulations — they have completed all 32 lessons of the B2 course!

Certificate of Completion

At the end of the course, students who complete the final feedback form and end-of-course CEFR assessment will receive a Certificate of Completion.

Use the certificate template provided. Click File —> Make a Copy, then you will be able to edit your own. Before printing or sending digitally, update the following fields and then send as a PDF:
- Number of hours completed
- Name of the course provider
- Student's full name

Distribute certificates promptly to recognize students' achievement.

Link to certificate template - https://docs.google.com/document/d/1drLQHfULCfOoN9HtrAZiu__WJC5wy_NZ/edit?usp=sharing&ouid=118059093709259566991&rtpof=true&sd=true

Master Vocabulary File – B2

UNIT 1

Lesson 1

- responsibilities
- contribute
- impact
- collaborate
- implement
- maintain
- streamline
- oversee
- deploy
- outcome

Lesson 2

- strength
- weakness
- achievement
- persistence
- adaptability
- leadership
- problem-solving
- initiative
- communication
- resilience

Lesson 3

- frontend
- backend
- scalability
- performance
- usability
- maintainability
- trade-off
- versatility
- complexity
- reliability

Lesson 4

- portfolio
- highlight
- relevance
- tailor
- achievement*
- flow
- concise
- confidence
- engage
- Q&A (questions and answers)

UNIT 2

Lesson 5

- consensus
- compromise
- justify
- perspective
- feedback
- constructive
- diplomatic
- evaluate
- alternative
- summarize

Lesson 6

- feedback*
- constructive*
- defensive
- framework
- actionable
- tone
- clarity
- empathy
- revision
- openness

Lesson 7

- tone*
- context
- clarity*
- recipient
- directness
- politeness
- medium
- miscommunication
- nuance
- professionalism

Lesson 8

- disagreement
- conflict
- escalate
- boundary
- compromise*
- assertive
- softening language
- resolve
- tone*
- empathy*

UNIT 3

Lesson 9

- step-by-step
- prerequisite
- sequence
- configuration
- dependency
- conditional
- troubleshoot
- draft
- clarity*
- formatting

Lesson 10

- function
- input
- output
- argument
- syntax
- example
- concise*
- format
- return
- readability

Lesson 11

- plain English
- simplify
- jargon
- term
- clarify
- tone*
- explain
- feature
- troubleshoot*
- accessible

Lesson 12

- heading
- bullet point
- hierarchy
- screenshot
- code block
- diagram
- readability*
- scanability
- caption
- layout

UNIT 4

Lesson 13

- component
- module
- service
- endpoint
- database
- data flow
- backend*
- frontend*
- architecture
- interaction

Lesson 14

- simplify*
- clarify*
- audience
- avoid jargon
- plain English*
- analogy
- paraphrase
- assumption
- example*
- explanation

Lesson 15

- dependency*
- relationship
- conditional*
- sequence*
- trigger
- impact*
- cause
- effect
- scenario
- consequence

Lesson 16

- scalability*
- maintainability*
- trade-off*
- reliability*
- performance*
- security
- cost-effective
- flexible
- drawback
- reasoning

UNIT 5

Lesson 17

- backlog
- sprint
- assign
- estimate
- priority
- negotiate
- dependency*
- deadline
- capacity
- blocker

Lesson 18

- retrospective
- feedback*
- constructive*
- improvement
- reflection
- suggestion
- tone*
- outcome*
- progress
- challenge

Lesson 19

- facilitate
- decision
- agenda
- summarize*
- clarify*
- feedback*
- suggestion*
- participation
- consensus*
- time limit

Lesson 20

- tool
- track
- update
- assign*
- comment
- document
- workflow
- dashboard
- repository
- status

UNIT 6

Lesson 21

- bug
- reproduce
- intermittent
- stack trace
- symptom
- environment
- root cause
- workaround
- log
- regression

Lesson 22

- root cause*
- chain of events
- error log
- trigger*
- malfunction
- hypothesis
- correlation
- misconfiguration
- dependency*
- fix / patch

Lesson 23

- optimization
- latency
- performance*
- resource usage
- load time
- scalability*
- cache
- throughput
- reliability*
- trade-off*

Lesson 24

- fix*
- patch
- regression*
- reproduce*
- investigate
- root cause*
- apply a fix
- verify
- log output
- rollout

UNIT 7

Lesson 25

- achievement*
- challenge*
- deadline*
- impact*
- initiative*
- adaptability*
- collaboration
- framework*
- mentorship
- reflection*

Lesson 26

- strength*
- weakness*
- initiative*
- resilience*
- adaptability*
- attention to detail
- accountability
- overcommit
- growth mindset
- constructive*

Lesson 27

- hypothetical
- scenario
- constraint
- trade-off*
- priority*
- escalate*
- allocate
- mitigation
- root cause*
- outcome*

Lesson 28

- milestone
- progress*
- long-term
- short-term
- aspiration
- competence
- initiative*
- evolve
- roadmap
- mentor*

UNIT 8

Lesson 29

- scope
- relevance*
- audience*
- overview
- section
- rationale
- clarify*
- emphasize
- objective
- summary

Lesson 30

- transition
- visual hierarchy
- readability*
- diagram*
- chart
- highlight*
- pace
- timing
- emphasis*
- summary*

Lesson 31

- delivery*
- flow*
- clarity*
- tone*
- confidence*
- gesture
- eye contact
- pacing
- posture
- revision*

Lesson 32

Review