

Managing Complex Tasks with AI (ChatGPT Canvas, Projects, Codex)

Author: Zsolt Lengyel, iTStudy Hungary Training and Research Centre

Reviewer: Mária Hartyányi, iTStudy Hungary Training and Research Centre

Learning objective

Planning a teaching unit, developing a course, building a question bank or preparing project documentation are not one-prompt tasks. The aim of this material is to show how advanced AI environments can serve as structured workspaces for such extended professional workflows. By the end of the material, teachers will be able to break a complex task into manageable stages with human checkpoints, use Projects to keep files, instructions and conversations together, edit longer texts and code iteratively in writing and code blocks (the successors of Canvas), understand what an agent such as Codex can do, and keep control over aims, quality and final decisions throughout.

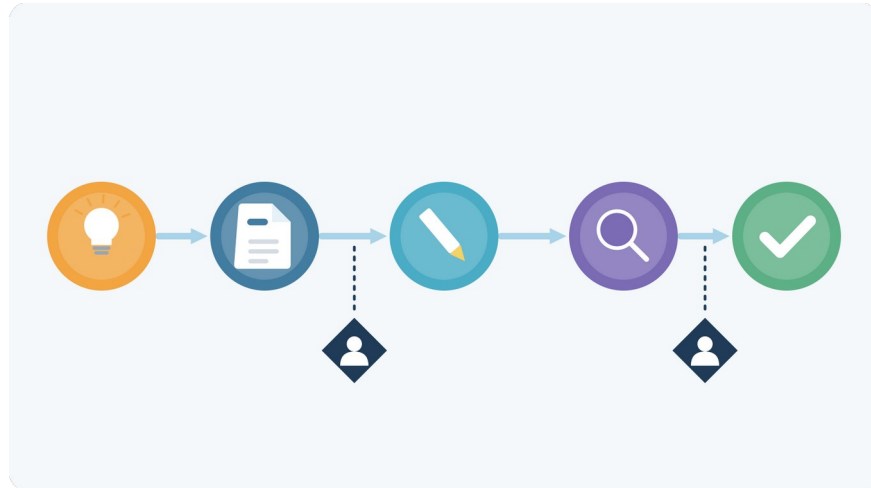
Introduction: from one-off prompts to workflows

Most people start using AI with isolated questions: one prompt, one answer. This works well for short tasks, but it breaks down when the work is longer, has several parts or continues over days. The conversation grows too long, earlier decisions are forgotten or compressed, files are scattered across chats, and it becomes unclear which version is the latest.

AI providers have responded with environments designed for longer work: project spaces that keep related material together, editors in which a document can be revised step by step, and agents that can carry out multi-step tasks on a computer. These tools are powerful, but they do not replace the teacher's role. The teacher defines the aims, sets the quality criteria, checks the intermediate results and makes the final decisions.

Breaking a complex task into stages

The most important technique is not tool-specific: split the work into stages and insert human checkpoints between them. A typical sequence for developing teaching material is:



Plan, draft, revise, check, finalise – with human checkpoints between the stages

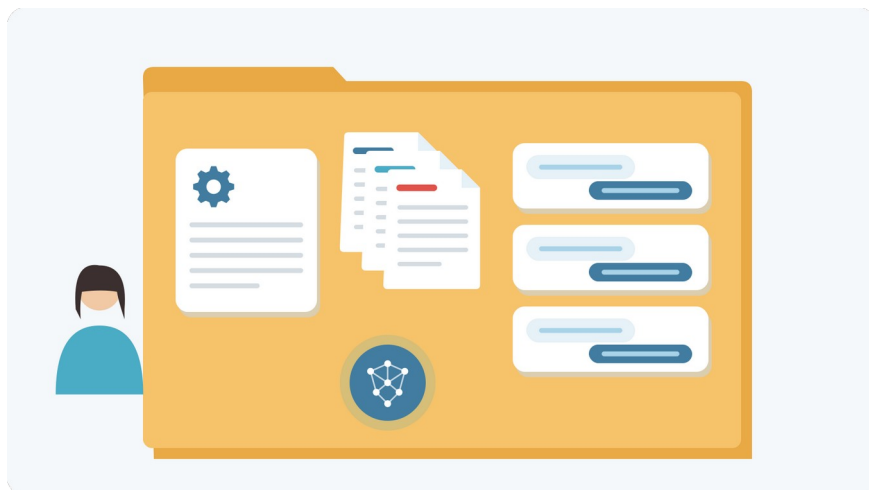
1. **Plan** – clarify aims, audience, scope and criteria; ask the AI to propose a structure and ask you clarifying questions.
2. **Draft** – generate the parts one by one, not everything at once.
3. **Revise** – improve specific sections with targeted instructions (“rewrite only section 3 for a lower reading level”).
4. **Check** – verify facts, safety content, level and alignment with the curriculum; ask the AI to critique its own draft against your criteria, then check it yourself.
5. **Finalise** – format, export, and store the final version together with the prompts that produced it.

Example: a staged workflow for a teaching unit

Stage 1: “Here is the curriculum excerpt (attached). Propose a 4-week unit plan for second-year HVAC apprentices: weekly aims, activities, assessment. Ask me questions first.” → *Checkpoint: the teacher approves or modifies the plan.*
 Stage 2: “Using the approved plan, draft the worksheet for week 1 only.” → Stage 3: targeted revisions → *Checkpoint: technical and safety check by the teacher.*
 Stage 4: “Create 10 quiz questions for weeks 1–2 with answer key, based only on the approved worksheets.”

Projects: a workspace for related work

A **Project** in ChatGPT is a smart folder that keeps related chats, files and instructions together, so that the assistant can rely on consistent context in every conversation of the project. Similar functions exist in other assistants (for example Projects in Claude or Gems and notebooks in Google’s tools).



A Project keeps instructions, reference files and related conversations in one place

- **Project instructions** tell the assistant how to behave within this project only (e.g. “You help me develop materials for the automotive mechatronics course. Always use British English terminology, include safety notes, and produce tables with answer keys.”). They take precedence over general custom instructions.
- **Files** – such as the syllabus, assessment criteria or a style guide – are available to every chat in the project. The number of files depends on the plan (at the time of writing: 5 in Free, 25 in Go/Plus, 40 in Edu, Pro, Business and Enterprise).
- **Memory** can be set to default or **project-only**. With project-only memory, chats cannot draw on conversations outside the project – useful for keeping courses or confidential work separate. Shared projects automatically use project-only memory.
- **Sharing:** projects can be shared with colleagues, who can chat in or edit the project depending on their role – a practical basis for joint course development.

A good rule of thumb is one Project per course or unit. Start each new sub-task in a new chat inside the Project: the chats stay short and focused, while the shared instructions and files ensure continuity.

From Canvas to writing and code blocks

In 2024 OpenAI introduced **Canvas**, a side-panel editor in which the user and ChatGPT could work on the same document or code, instead of regenerating whole answers. In May 2026 Canvas was withdrawn from the current GPT-5.5 models; its functions now appear directly in the chat as **writing blocks** and **code blocks**. Longer texts can be opened in a focused full-screen editor, edited, copied and downloaded; code blocks can preview or run their content. Other assistants offer comparable workspaces (for example Artifacts in Claude or Canvas in Gemini).



Editable writing blocks and runnable code blocks appear directly in the conversation

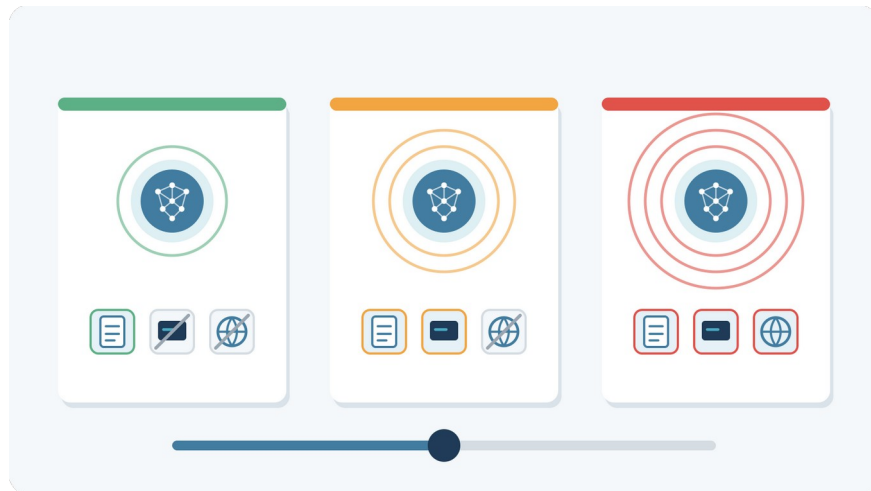
Whatever the tool is called, the working method is the same:

- Ask for a first draft, then edit it in place – yourself or with targeted requests for selected parts.
- Change one thing at a time (“shorten the introduction”, “add a safety note after step 4”), so you can see and check the effect.
- Keep versions: download or copy the approved version before asking for major changes.
- Use code blocks for small interactive elements, such as a self-check quiz or a calculator for a workshop formula, and test them before use.

Codex: an agent that works on the computer

Codex is OpenAI's AI agent originally built for writing, reviewing and running code. It is available in the ChatGPT plans (with usage limits depending on the plan) in the ChatGPT desktop app, on the web, in the command line and in code editors. Unlike a chatbot, an agent does not only answer: it plans steps and executes them – it reads and writes files, runs programs, and can even use a browser or other applications. Comparable agents include Claude Code and Claude Cowork.

Teachers do not need to be programmers to benefit from this. Typical educational uses include converting a Word document into an interactive web page, generating quizzes with scoring, batch-renaming or reformatting dozens of files, or building a small tool for a lesson. The interactive learning materials of the VITA project itself are produced this way: Codex receives the Word documents of a micro-learning unit and generates a paginated, illustrated web module with a quiz.



The user decides how much freedom the agent has: it asks before every step, proposes changes for review, or has full access

Because an agent acts, clear instructions and appropriate permissions are essential:

- Choose the permission level consciously. Start with the mode in which the agent asks for approval, observe how it works, and only then give it more freedom.
- Let it work in a separate folder with copies of your files – never on the only copy of important material.
- State clearly whether you want exploration or action. If you only want an analysis, say so explicitly, otherwise the agent may start modifying files.
- Watch its reasoning while it works; if it goes in the wrong direction, stop it and correct the instruction.

Keeping the teacher in control

Checklist for extended AI workflows

- Define aims, audience and quality criteria before you start – and write them into the project instructions.
- Insert checkpoints: approve the plan before drafting, and check facts and safety content before finalising.
- Keep the originals and the approved versions; record the key prompts.
- Never upload learners' personal data; use institutional accounts where available.
- The final decision – and the professional responsibility – remains with the teacher.

Takeaways for teachers

- Complex tasks need a workflow, not a single prompt: stage the work and review between stages.
- Use one Project per course or unit, with instructions and reference files; start new sub-tasks in new chats.
- Edit long texts and code iteratively in writing and code blocks, one change at a time, and keep versions.
- Agents such as Codex can carry out multi-step tasks on the computer – give them clear instructions, limited permissions and copies of files.
- AI accelerates the work; aims, quality and final decisions stay with the teacher.

References

- OpenAI Help Center (2026): Projects in ChatGPT; Using Codex with your ChatGPT plan; ChatGPT release notes (28 May 2026: Canvas replaced by writing and code blocks).
- OpenAI (2026): Introducing the Codex app.
- Lengyel, Zs. (2026): Introduction to the concepts of artificial intelligence; Concrete applications for content generation. VITA training presentations.

Links

- Projects in ChatGPT: help.openai.com/en/articles/10169521-projects-in-chatgpt
- Using Codex with your ChatGPT plan: help.openai.com/en/articles/11369540
- ChatGPT release notes: help.openai.com/en/articles/6825453-chatgpt-release-notes

Copyright © 2026 VITA Consortium

This work may be reused and adapted under the terms of the Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).



Funded by the European Union. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the National Agency.