Work effectively in the GitHub Copilot app

Session M7-01 | Product and Delivery Teams | Beginner

Work in the GitHub Copilot app

Work in the GitHub Copilot app with repository context. Save work that others can review.

Delivery Decision Studio is the six-session path. **Service Request Portal** is the initiative.

Save work beyond the chat

Use the chat to work through an idea. Keep project context, source references, issues, and saved work available for review after the conversation ends.

```
Gather evidence → form a proposal → review it → save the decision → verify it
```

Check Copilot's explanation against the source facts.

Context has a cost

Context choice	When it helps
One named file	The answer depends on a specific rule or API
A project or repository	The task needs relationships across files
An issue or pull request	The task needs delivery history and discussion
No repository context	The work is general and contains no project facts

Tell Copilot why each source matters. **Do not make it guess.**

Separate facts, assumptions, and decisions

Type	Example
Fact	"Issue 42 names a response-time problem."
Assumption	"The mobile client uses the documented endpoint."
Decision	"The product owner will confirm the client behavior."

Separating these categories makes uncertainty visible.

Know the main work areas

Area	Use it for
Chats	Exploration and interviews
Projects	Repository-grounded work
My work	Issues, pull requests, and sessions
Search	Finding repository information
Automations	Recurring or on-demand tasks
Customize	Skills, agents, MCP servers, and canvases

Context comes before the prompt

Use only permitted sources. Attach the smallest set that answers the question.

Name the files, issues, or repository that should support the answer.

Choose the session mode

Mode	Use in this path
Interactive	Questions, interviews, and reviewed drafts
Plan	Review a multi-step approach before changes
Autopilot	Work with agreed scope and permission to act without review of every action

Session M7-01 uses Interactive, then Plan. The learner reviews every write.

Use context controls

- `@` adds a file.
- `#` references an issue.
- `/` opens commands.
- `/context` shows current context usage.
- `/compact` summarizes a long active session.
- `/clear` or `/reset` starts fresh.

Ask for sources

```
Explain the initiative for a product owner.
Cite the supplied source for each factual claim.
Mark missing information as unknown.
```

A confident answer without a source is still unverified.

Inspect what Copilot can use

- Skills provide repeatable procedures.
- Agents carry role-specific instructions and tools.
- MCP servers connect Copilot to systems.
- Plugins install reusable packages that can include MCP servers and other customizations.
- Canvases provide visible shared state.
- Automations repeat tasks with a defined scope.

Availability depends on the plan, policy, and installation.

Use one repeatable work pattern

```
Read -> propose -> review -> write -> verify
```

Every later Module 7 session uses this pattern.

Save the working brief

The lab creates `initiative-context-brief.md` with:

- facts and sources;
- unknowns;
- useful GitHub artifacts;
- the next decision and owner.

The brief must make sense without the chat history.

Revise the brief when sources change

```
Initial source
  → first brief
  → approved late evidence
  → focused revision
  → fresh-session verification
```

Keep the brief consistent with source information the owner has approved.

Lab

1. Open the repository as a Copilot project.
2. Inspect the available capabilities.
3. Start fresh without the source, then compare an unscoped answer with a grounded answer.
4. Create and commit the initiative context brief.
5. Apply approved late evidence.
6. Review and commit the focused revision.
7. Ask `/rubber-duck` for independent critique.
8. Check the brief in a fresh session.

Stop if the app or repository context is unavailable.