Codebook Structure: Theme → Subtheme → Code

Theme 1: Understanding and Localization

Programmers Prioritize Comprehension for Long-Term Projects

Subtheme 1.1: Code Comprehension
Summaries Help Understand Code
Reading Summaries Takes Effort

Subtheme 1.2: Progressive Narrowing
Summaries Help Progressively Narrowing
Summaries Help Determine If Intended Location

Theme 2: Specifying Changes

Subtheme 2.1: Direct Instruction Prompting
Direct Prompting Is Flexible
Direct Prompting Prone to Errors
Direct Prompting Enables Confident Specification

Subtheme 2.2: Summary-Mediated Prompting
Editing Summaries Reduces Typing Effort
Summaries Provide Accurate Technical Vocabulary
Summaries Help Scaffold Complex Prompts
Summaries Help Prevent Unintended Edits
Summaries Force Comprehension Before Modification
Summaries Provide a More Comprehensive Prompt

Theme 3: Validating Output

Code Execution Preferred for Validation
Early Comprehension Prevents Future Debugging Costs
Comprehension Is Necessary For Validation
Summaries Support Semantic Validation
Lack of Familiarity Hinders Validation

Theme 4: Prompting Strategy Choice

Prompting Strategy Switching Based on Context

Subtheme 4.1: Goal-Driven Strategy
Urgency Favors Direct Prompting
Maintenance Interest Favors Summary Prompting

Learning Motivates Summary Prompting

Subtheme 4.2: Task Conditions
Familiar Code Favors Direct Prompting
Unfamiliar Code Encourages Summary Use
Simple Tasks Favor Direct Prompting
Complex Tasks Trigger Summary-Mediated Prompting
Large Scope Favors Direct Prompting
Small Scope Favors Summary Prompting
Clear Intentions Favor Direct Prompting
Trust In LLMs Favors Direct Prompting
Vague Intentions Lead to Summary Use

Theme 5: Usability of Summaries

Overly High-Level Summaries Obscure Edit Points
Overly Detailed Summaries Negatively Impact Usefulness
Appropriate Granularity Is Important for Editing
Summary Usefulness Depends on Goal Alignment
Structured Summaries Aid Readability
Expandable Summaries Improve Usability
Summary-Code Mapping Improves Traceability
Inconsistent Summaries Undermine Usability
Intermediate Steps Before Committing

Theme 6: Challenges of LLM Usages

LLM Modifications May Introduce Unintended Effects
Unfamiliar Code Encourages LLM Editing
Difficult To Determine Correctness
Locating Where to Modify Is Challenging
Developers Prefer Incremental Changes
Describing Bug Context To LLMs Is Effortful
Describing Code Dependency To LLMs Is Effortful
Describing Runtime Behavior To LLMs Is Effortful
Describing Vague Intentions To LLMs Is Effortful
Over-Specification Leads to Unintended Effects
Iterative Correction Increases Cognitive Burden
Tool Performance Affects Prompting Experience

Theme 7: Industrial Impacts

Safe Integration Requires Comprehension
Collaboration Requires Understanding Others' Code

LLMs Lack Internal Company Code Context
Unit Tests Support LLM Validation