

Format

Pattern	Example
type(scope): summary	frame(program): redefine current problem state for AI-era redesign

Commit Types

Type	Use When	Typical Examples
frame	The governing understanding changes	problem reframing, scope clarification, correcting a higher-order assumption
build	Creating substantial new work	new artifacts, new sections, new matrices, new program/course content
refactor	Structure improves without major change to core intent	reorganizing documents, tightening structure, improving sequence or alignment
synth	Combining multiple artifacts into a higher-order output	merging sections into a full draft, combining several artifacts into one summary
review	A change is driven mainly by feedback or inspection	supervisor revisions, advisory revisions, review-response changes
meta	Process or repo guidance changes	workflow notes, commit taxonomy, README/process guidance
chore	Light maintenance only	renaming, moving files, formatting cleanup

Default Scopes

Scope	Use For
program	program design, transition models, bridge/successor work
course	course-level artifacts and redesign work
whitepaper	white paper sections, appendices, full draft work
case-study	case studies and reflections
workflow	process conventions, commit taxonomy, operational guidance
repo	repo-level organization and maintenance

Other scopes are allowed. Consistency is the key point.

Sample Commits

Situation	Sample Commit
redefine the core problem	<code>frame(program): redefine bridge and successor relationship</code>
add new course artifact work	<code>build(course): add Python bridge implementation patterns</code>
tighten existing program structure	<code>refactor(program): tighten semester-by-semester bridge structure</code>
merge paper sections into one draft	<code>synth(whitepaper): merge sections and appendices into full draft</code>
revise after supervisor feedback	<code>review(whitepaper): revise framing for supervisor readability</code>
update the commit system itself	<code>meta(workflow): simplify commit taxonomy for artifact ecosystems</code>
rename or reorganize files	<code>chore(repo): rename and reorganize support files</code>

Keep It Simple

Rule	Reminder
------	----------



Keep It Simple (cont)

- 1 Prefer consistency over perfect precision
- 2 Use broad scopes, not overly specific ones
- 3 Avoid inventing new types unless needed repeatedly
- 4 If a commit spans multiple artifacts, classify by the dominant action

Quick Decision Rule

If the main move is ...	Use
changing the governing meaning	frame
creating substantial new material	build
reorganizing existing structure	refactor
combining multiple strands	synth
revising from feedback	review
updating process/repo guidance	meta
doing minor maintenance only	chore

RBA Definition

RBA (*Refraction-Based Architecture*) is a way of using AI to develop complex work through structured refraction rather than one-shot generation, so that meaning, architecture, and artifact coherence are preserved as the work evolves.

It is a governed human-AI collaborative process in which intent is progressively refracted across multiple artifact layers while preserving coherence, allowing lower-level work to inform and refactor higher-level understanding when reality reveals hidden constraints, opportunities, or structural mismatches.

(*Collaboration* was intentionally **deferred** because it semantically suggests two sentient beings working together, which risks anthropomorphizing AI in ways that are conceptually misleading. **Refraction** is more appropriate because, like light passing through a medium and changing direction or separating into visible components, it captures how human intent is perceived, transformed, decomposed, and delegated across different layers of artifacts, constraints, and AI-mediated processes.)



By Cash (CashM)
cheatography.com/cashm/

Not published yet.
 Last updated 26th April, 2026.
 Page 3 of 3.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

