Test-Time Tool Chain Evolve

Atomic tool | **Code-wrapped Atomic Tool**

```python
def get_project_info(path):
    resp = gitlab_api(endpoint=f"/projects/{path}")
    return {"name": resp["name"], "stars":resp["star_count"],
            "forks":resp["forks_count"]
    }
```

```python
def get_contributors(path):
    resp = gitlab_api(endpoint=f"/projects/{path}/contributors")
    return {"total": len(resp), "top_5": resp[:5]
    }
```

```python
def get_commits(path):
    resp = gitlab_api(endpoint=f"/projects/{path}/commits")
    return {"count": len(resp), "recent": resp[:20],
        "authors": unique_authors(resp),
    }
```

```python
def get_branches(path):
    resp = gitlab_api(endpoint=f"/projects/{path}/branches")
    protected = [b for b in resp if b["protected"]]
    return {"total": len(resp),
        "protected": len(protected),
        "default": resp[0]["name"],
    }
```
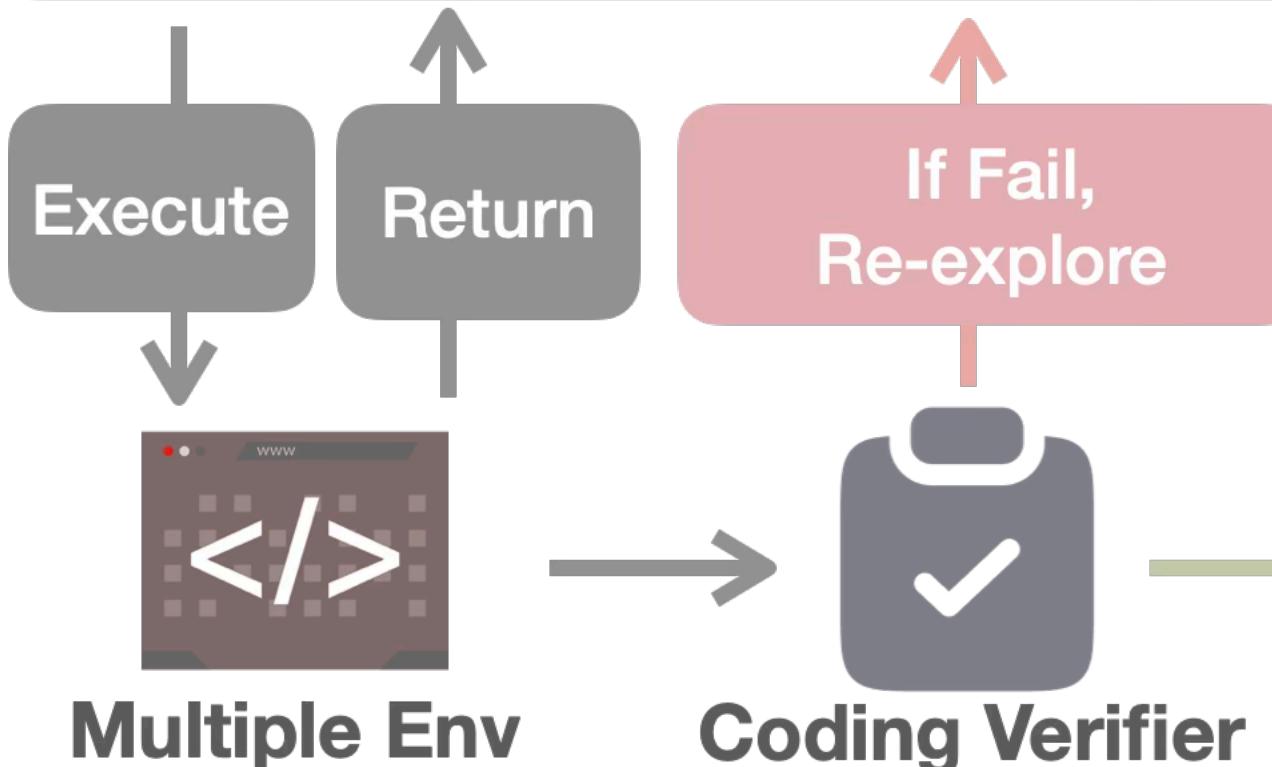
Task:
Analyze multiple gitlab repositories

Tool Chain Execution

Sample Task | **Task Library**

Iterative Skill Composition

Skill

```python
def gitlab_project_analyzer(path):
  1  info = get_project_info(path)
  2  contributors = get_contributors(path)
  3  commits = get_commits(path)
  4  branches = get_branches(path)

    return {
        "path": project_path,
        "name": info["name"],
        "stars": info["stars"],
        "contributors":
contributors["total"],
        "recent_commits":
commits["count"],
        "branches": branches["total"],
    }
```

Execute | **Return** | **If Fail, Re-explore**

Multiple Env | **Coding Verifier**

If Pass, Save new skill

Skill Library

- region_earthquakes_analyzer
- vocabulary_card_builder
- **gitlab_project_analyzer**
- country_economic_analyzer
- tvmaze_show_data_extractor
- city_weather_analyzer
- cocktail_menu_generator
- pokemon_info_collector

Usable Skill retrieval