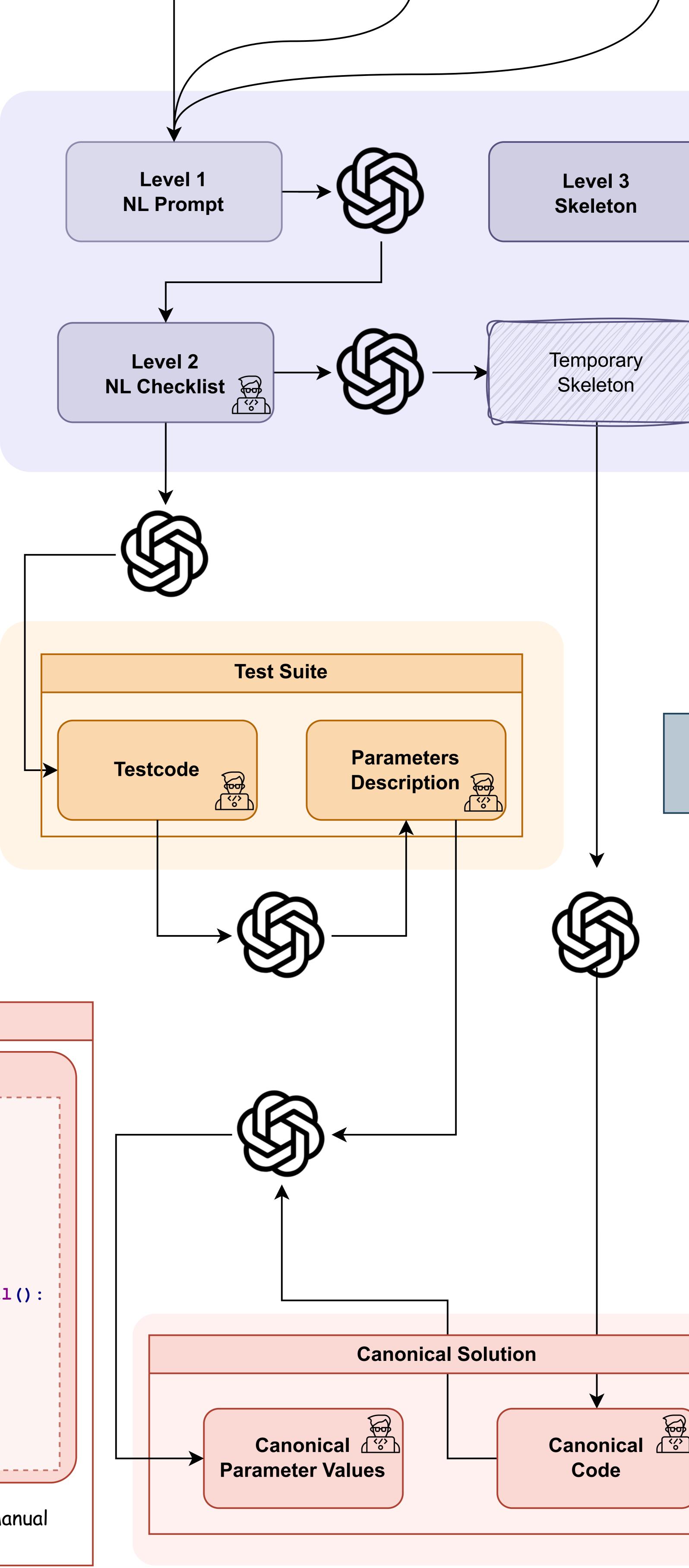
Input Level 1 NL Prompt - Source: MetaGPT, AGILE C DER and original. Create a website for a Todo application, additional chart for analysis will be better. Level 2 NL Checklist - Generated by GPT-40 with manual reviewing "Page-Todo List": "Create New List": "Allows users to create a new Todo list", "View Tasks": "Displays all tasks under a selected list", ... Level 3 Skeleton - Generated by Masker module from Canonical Solution class TodoListView(ListView): def get_queryset(self): """ Gets the list of TodoLists, updating their completion status. """ pass

pass **Test Suite** Selenium Testcode - For Website mission di def test_view_tasks(driver, test_url, todo_list_selector_class_name): driver.get(test url) todo_list_selector = driver.find_element(By.CLASS_NAME, todo_list_selector_class_name) todo list selector.click() assert 'New List' in driver.title Parameter "test url": "the url for test" Descriptions "todo_list_selector_class_name": "the class name for a todo list selector" Console&File Testcode - For No-Ul mission Parameters' values will be reasoned and def test move_player(direction, expected_output): answered by agent stdout = subprocess.send command("cd "+direction) based on their assert utils.string_similarity(expected_output, stdout) description and given solution Parameter "direction": "A destination name..." Descriptions "expected_output": "...player input a wrong selection..."



Natural Language (NL) Prompt Source

MetaGPT

ProjectDev

AGILE C DER

SoftwareDev

Legends

Human

Reviewing

LLM

Reasoning

ProjectEval

Original

Masker

Canonical Solution

Canonical Parameter Values: the standard answer correspond to Canonical Solution for the testcase.

Note: every solution will have their own parameter values to getpass the testcase

Canonical Parameter Values

"test_url":
 "http://localhost:8000/todos/"
"todo_list_selector_class_name":
 "todo-list-selector"

Canonical Parameter Values

"direction": "home"
"expected_output":
 "You are now in \\home\n"

Canonical Code # example app/views.py class TodoListView (ListView): template name = 'todo list.html' model = TodoList def get queryset(self): queryset = TodoList.objects.all() for todolist in queryset: flag = True for todoitem in todolist.todoitem set.all(): if todoitem.completed: continue flag = False break todolist.completed = flag todolist.save() return queryset

Canonical Code: A reference solution generated by LLM with manual reviewing passable to all the testcase