

# Agent Improvements Documentation

## Overview

This document details the comprehensive improvements made to the Teach-Genie agent system to achieve **consistent, expert-level curriculum generation** that is:

- Level-appropriate (School → Professional)
- Duration-aware (15 → 120+ minutes)
- Globally accessible and culturally neutral

## 1. Utils Module (`app/agents/utils.py`)

### New Additions

#### LEVEL\_PROFILES Dictionary

A comprehensive profile system defining characteristics for each educational level:

```
LEVEL_PROFILES = {
    "School": {
        "age_range": "13-18 years",
        "depth": "Foundational understanding with concrete examples",
        "vocabulary": "Simple, everyday language; avoid jargon",
        "prerequisites": "Basic literacy and numeracy",
        "examples": "Relatable scenarios from daily life, school, family",
        "complexity": "Single-step reasoning, clear cause-effect",
        "assessment": "Recognition and recall, basic application",
        "cognitive_load": "Low - one concept at a time",
        "global_context": "Universal examples accessible across cultures"
    },
    "Undergraduate": {...},
    "Postgraduate": {...},
    "Professional": {...}
}
```

### Helper Functions

#### Function Purpose

`get_level_profile(level)` Returns the complete profile dict for a given level

```
get_level_guidance(level) Generates formatted guidance string for prompts
```

## Enhanced `duration\_profile()` Function

### Before:

```
def duration_profile(minutes: int) -> Dict[str, Any]:  
    return {  
        "sections": 4,  
        "depth": "moderate",  
        "examples": 2,  
        "quiz_questions": 5,  
        "resources": 4  
    }
```

### After:

```
def duration_profile(minutes: int) -> Dict[str, Any]:  
    # Includes new fields:  
    # - depth_guidance: Specific instructions for content depth  
    # - pacing: Minutes per section guideline  
    # Dynamic scaling from ultra-short (15min) to comprehensive (120+min)
```

## Why It's Better

**Centralized Configuration:** All level-specific parameters in one place

**Consistency:** Every agent uses the same level definitions

**Maintainability:** Easy to add new levels or modify existing ones

**Rich Context:** Provides vocabulary, complexity, and cultural guidelines

## 2. Planner Agent (`app/agents/planner.py`)

### Key Changes

#### System Prompt Enhancement

##### Before:

```
"You are an expert curriculum designer. Return JSON only."
```

##### After:

```
"You are a world-class curriculum architect with expertise in educational psychology, instructional design, and learning science. You design curricula used by educators globally, ensuring content is culturally neutral, pedagogically sound, and appropriate for the target audience level."
```

## User Prompt Improvements

### Aspect Before After

Level Context Just level name Full profile with age range, depth, vocabulary

Duration Handling Basic section count Pacing guidance, depth instructions

Section Structure Generic list Detailed structure with timing, goals, engagement hooks

Global Standards Not addressed Explicit cultural neutrality requirements

## New Prompt Sections

### Educational Context Block

- Target audience characteristics
- Prior knowledge assumptions
- Vocabulary constraints

### Curriculum Design Principles

- Progressive complexity
- Active learning integration
- Universal accessibility

### Section Structure Template

- Learning goal per section
- Engagement hooks
- Time allocation
- Transition logic

## Why It's Better

**Expert-Level Output:** Prompts emulate professional curriculum designers

**Pedagogically Sound:** Follows learning science principles

**Globally Acceptable:** Explicit instructions for cultural neutrality

**Structured Planning:** Each section has clear purpose and timing

## 3. Content Agent (`app/agents/content.py`)

## Key Changes

### Level-Aware Content Generation

#### Before:

```
user_prompt = f"""
Topic: {topic}
Level: {level}
Section: {section_title}
...
"""
```

#### After:

```
level_profile = get_level_profile(level)
user_prompt = f"""
TOPIC: {topic}
EDUCATION LEVEL: {level}
TARGET AUDIENCE: {level_profile['age_range']}
```

LEVEL-SPECIFIC REQUIREMENTS:

- Vocabulary: {level\_profile['vocabulary']}
- Depth: {level\_profile['depth']}
- Complexity: {level\_profile['complexity']}
- Examples: {level\_profile['examples']}
- Global Context: {level\_profile['global\_context']}

```
...
"""
```

### Enhanced Resource Generation

#### Before: Generic 4-6 resources

#### After:

```
RESOURCES (exactly 8 items):
Generate a balanced mix across these categories:
- 2 theoretical/foundational resources
- 2 practical/applied resources
- 2 video/multimedia resources
- 2 reference/documentation resources

Each resource with:
- Type categorization (video, article, course, documentation, tool, book)
- Difficulty indicator (matches level)
- Brief description of value
```

### Global Accessibility Guidelines

#### New section added:

```
GLOBAL ACCESSIBILITY:
- Use metric units (km, kg, °C) as primary
- Reference international standards and organizations
- Include examples from multiple geographic regions
```

- Avoid culture-specific idioms or references
- Use globally recognized terminology

## Why It's Better

**Level-Appropriate Language:** Content matches audience vocabulary

**Richer Resources:** 8 balanced resources across categories

**Cultural Neutrality:** Explicit guidelines for global content

**Consistent Depth:** Matches level complexity requirements

## 4. Quiz Agent (`app/agents/quiz.py`)

### Key Changes

#### Complete Prompt Redesign

##### Before:

"You are an expert assessment designer creating multiple-choice questions."

##### After:

"You are an expert assessment designer with specialization in creating globally accessible, level-appropriate evaluations. Your questions are used by educational institutions worldwide and must be culturally neutral, pedagogically sound, and aligned with international educational standards."

#### Level-Appropriate Question Design

##### Level Question Characteristics

School Simple scenarios, single-concept focus, clear correct answers

Undergraduate Applied scenarios, multi-step reasoning, some ambiguity

Postgraduate Complex scenarios, synthesis required, nuanced options

Professional Real-world cases, decision-making focus, industry context

#### New Prompt Sections

#### Question Design Principles

- Progressive difficulty within quiz
- Cover different cognitive levels
- Real-world scenario focus
- Clear, unambiguous wording

### Global Accessibility for Questions

- Use universally recognized scenarios
- Avoid culturally specific references
- Use metric units and international standards
- Names should be culturally diverse
- Avoid region-specific regulations

### Option Design Guidelines

- All options must be plausible
- Correct answer shouldn't be obviously longer
- Avoid "all of the above" patterns
- Distractors based on common misconceptions

## Why It's Better

**Scenario-Based:** Questions test application, not memorization

**Globally Accessible:** Culturally neutral scenarios and examples

**Cognitively Appropriate:** Question complexity matches level

**Better Distractors:** Options based on real misconceptions

## 5. Key Takeaways Agent (`app/agents/key_takeaways.py`)

### Key Changes

#### Level Integration

**Before:** No level-specific handling

**After:**

```
from app.agents.utils import get_level_profile
level_profile = get_level_profile(level)

# Used in prompt:
# - Vocabulary matching level
# - Complexity appropriate to audience
# - Global context requirements
```

## Enhanced Prompt Structure

**Before:**

```
"Generate EXACTLY 5 key takeaways..."
```

**After:**

```
KEY TAKEAWAY DESIGN GUIDELINES:

1. STRUCTURE (EXACTLY 5 takeaways):
- "title": Short header (2-5 words) - the key concept
- "description": One complete sentence explaining the insight

2. CONTENT QUALITY:
- Focus on "aha moments" and core insights
- Make them memorable and quotable
- Ensure practical applicability
- DO NOT repeat learning objectives verbatim

3. GLOBAL ACCESSIBILITY:
- Use universally understood concepts
- Avoid culture-specific references
- Include globally relevant examples
```

## Improved Examples

**Before:**

```
{
  "title": "Causes of Conflict",
  "description": "Economic inequality..."
}
```

**After:**

```
[
{
  "title": "Supply Meets Demand",
  "description": "Market prices naturally adjust to balance..."
},
{
  "title": "Code Reusability",
  "description": "Functions allow you to write logic once..."
}]
```

## Better Fallback Content

**Before:**

```
{"title": f"Introduction to {topic}", "description": f"{topic} is a fundamental concept..."}
```

**After:**

```
{"title": "Core Concept", "description": f"{topic} represents a foundational idea with applications across multiple domains."}  
 {"title": "Global Relevance", "description": f"{topic} plays a significant role in international contexts and cross-cultural applications."}
```

## Why It's Better

**Insight-Focused:** Takeaways are "aha moments", not objectives

**Memorable:** Short titles, complete explanations

**Level-Appropriate:** Language matches audience

**Globally Relevant:** Universal applicability emphasized

## Comparison Summary

Aspect	Before	After
Level Awareness	Basic level name only	Comprehensive profiles with 9 parameters
Duration Handling	Fixed section counts	Dynamic scaling with depth guidance
Global Accessibility	Not addressed	Explicit in every agent's prompts
Vocabulary Control	None	Level-specific vocabulary guidelines
Question Quality	Generic MCQs	Scenario-based, cognitively appropriate
Resource Diversity	Random selection	Categorized, balanced distribution
Content Depth	One-size-fits-all	Scaled to level complexity
Cultural Neutrality	Not considered	Metric units, diverse names, universal examples

## Impact on Generated Content

### School Level (13-18 years)

- Simple vocabulary, concrete examples
- Daily life scenarios

- Single-concept questions
- Foundational takeaways

## **Undergraduate Level (18-24 years)**

- Academic vocabulary with definitions
- Case studies and research references
- Applied reasoning questions
- Conceptual connection takeaways

## **Postgraduate Level (22-30 years)**

- Technical terminology expected
- Research methodology integration
- Complex analytical questions
- Advanced synthesis takeaways

## **Professional Level (25+ years)**

- Industry-standard terminology
- Real-world case studies
- Decision-making questions
- Practical implementation takeaways

## **Files Modified**

### File Changes

`app/agents/utils.py` Added LEVEL\_PROFILES, get\_level\_profile(), get\_level\_guidance(), enhanced duration\_profile()

`app/agents/planner.py` Complete prompt redesign with expert curriculum design approach

`app/agents/content.py` Level-aware content generation, enhanced resources, global guidelines

`app/agents/quiz.py` Scenario-based questions, level-appropriate complexity, cultural neutrality

`app/agents/key_takeaways.py` Level integration, improved prompt structure, better fallbacks

# Testing Recommendations

**Level Comparison Test:** Generate lessons on the same topic at all 4 levels

**Duration Test:** Generate lessons at 15, 30, 60, and 90 minutes

**Cultural Neutrality Audit:** Review content for region-specific references

**Vocabulary Analysis:** Verify language complexity matches level

**Question Quality Review:** Assess scenario relevance and option quality

*Document created by: TeachGenie Development team*

*Version: 2.0 - Expert Curriculum Generation*