📄 FSD_04 – Dynamic Qualification Generator

🧠 Purpose:

This module is responsible for generating intelligent, personalized follow-up questions **automatically**, based on:

- Gaps found in discovery tag coverage (from **FSD_03**)
- New tags inferred from documents that were **not addressed** in onboarding
- Inconsistencies or weak confidence zones that need clarification
- Cross-referenced business-critical categories not confirmed yet

This is where Audira acts like a **proactive consultant**, asking the right questions — not generically, but **targeted to each SMB's real gaps**.

🧩 FSD_04 – Section Breakdown

Section	Description
1. Scope	What this module does and what it hands off to the agent
2. Input Requirements	The tag gap map, onboarding answers, segment anchors
3. Prompt Generation Logic	How follow-up questions are composed
4. Contextual Boosting	How previous answers or segments shape the new question
5. Output Format	Standard structure for storing, displaying, and linking questions
6. AI Runtime Behavior	When and how these questions are shown (UI/agent timing)
7. Edge Handling	What happens if 0 or >15 questions are detected
8. Future Enhancements	Adaptive learning, multilingual refinements, agent tuning

---

Section 1: Scope

🎯 Purpose:

The **Dynamic Qualification Generator** creates up to 15 smart follow-up questions per client, customized to:

- Fill **unanswered or weak discovery tags**
- Clarify **conflicting information**
- Explore **new insights discovered in uploaded files**
- Validate **missing business-critical categories**

It ensures that Audira doesn't just "listen" — it actively **thinks like a consultant**, guiding SMBs to fully define their business before agent deployment.

✅ Responsibilities:

1. **Question Composition**
   Dynamically generates natural language questions using AI prompt templates and rules. Every question is tied to a **discovery tag**.
2. **Context Injection**
   If a file segment or onboarding answer partially covered a tag, the question adds that **reference in the prompt**.
   e.g. "We found a payout policy, but not the schedule. Can you clarify?"
3. **Tag-Based Tracking**
   Every generated question is tagged with:
   - The discovery tag it relates to
   - Priority (core, advanced, optional)
   - Source (gap, ambiguity, new insight)
4. **Limit Control & Flexibility**
   Generates **up to 15 questions**, but dynamically adjusts:
   - Can create **less** if gaps are already covered
   - Can **exceed 15** if the system detects many new tags → activates a follow-up queue (optional)
5. **Feeds Into Agent Experience**
   Final questions are presented to the consumer in the last onboarding stage or routed via agent dashboard (depending on UI setting).

❌ Not in Scope:

- This module **does not display** the questions (UI controls that)
- It doesn't **score answers** — that happens in FSD_05: Readiness Validator
- It doesn't generate prompts for final agent usage — that belongs to the Prompt Chain Logic

---

📎 References:

- 📘 *AUDIRA DISCOVERY TAGS DICTIONARY* – defines question intent and templates
- 📘 *AUDIRA AGENT ONBOARDING FRAMEWORK* – shows which tags are already covered

- 📘 *AUDIRA PROMPT CHAIN & LLM LOGIC FLOW* – receives any clarification-based prompt adjustments
- 📘 *AUDIRA PRE-LAUNCH VALIDATOR SPEC* – tracks answered vs unanswered tag status

---

Section 2: Input Requirements

🎯 Purpose:

This module takes in the full semantic understanding built by previous layers to generate the most relevant questions possible. It depends on inputs from:

📄 A. Tag Coverage Map (from FSD_03)

For every discovery tag:

```
{
  "tag_id": "payout_policy",
  "status": "partially_covered",
  "best_confidence": 0.71,
  "sources": ["onboarding_q7"],
  "suggested_action": "trigger_question",
  "priority": "core"
}
```

Used to determine which tags require qualification.

📄 B. Enriched Content Segments (from FSD_02)

For context-aware prompting:

```
{
  "segment_id": "seg_45",
  "text": "Vendors are paid every 14 days.",
  "matched_tags": ["payout_policy"],
  "semantic_role": "policy_clause",
  "confidence": 0.68
}
```

Used to inject reference examples into the generated question when a tag is weakly covered.

C. Onboarding Answers (from AGENT ONBOARDING)

For tracking what the customer already declared:

```
{
  "question_id": "q_10",
  "answer_text": "We use Stripe to process all payments and split vendor
earnings automatically.",
  "linked_tags": ["payment_method", "payout_policy"]
}
```

Helps avoid re-asking confirmed facts, or tailor new questions to what's already been said.

D. Discovery Tags Dictionary (from TAG DICTIONARY)

Used to understand:

- Which tags are required vs optional
- Predefined keywords and phrases
- Suggested AI prompt templates per tag

```
{
  "tag_id": "revenue_model",
  "category": "business_model",
  "priority": "core",
  "template": "How does your business earn revenue? Is it based on
transactions, subscriptions, or another model?"
}
```

Summary Input Flow:

Input Source	Consumed For
`tag_coverage_map`	Gap analysis + question triggers
`segments`	Context-aware phrasing
`onboarding_answers`	Prior answers → avoid duplication
`discovery_tags_dictionary`	Template matching + metadata injection

Section 3: Prompt Generation Logic

🎯 Purpose:

To generate **natural language business questions** for each uncovered or weakly covered discovery tag — using pre-defined templates, adaptive context, and embedding-based tailoring.

🧠 Logic Flow Per Question:

1. Select Trigger Tags

Pull tags with:

- `status = uncovered` OR
- `status = partially_covered` AND `confidence < 0.85`

Respect the `priority = core` first, then fill in `advanced`, `optional`.

---

2. Fetch Prompt Template

From *DISCOVERY TAGS DICTIONARY*:

```
"template": "How does your business earn revenue? Is it based on
transactions, subscriptions, or another model?"
```

If not available, fallback to `LLM-based phrase synthesis` using tag description and keywords.

---

3. Context-Aware Injection (Optional)

If a low-confidence segment exists for that tag, reference it:

```
"context": "We noticed you mentioned vendors are paid every 14 days. Can you
explain if this is automated or manual?"
```

This creates **clarifying questions**, not cold ones.

---

4. LLM Boosting (Optional for Flex)

For more dynamic phrasing:

- Use prompt like:
- `Generate a clear, concise business question to clarify the vendor payout structure. Reference: 'Vendors are paid every 14 days via ACH.'`
- Outputs:

  "Is your vendor payout schedule fixed, or can it vary based on sales?"

---

5. De-duplication

- Avoid re-asking what was already answered
- Don't trigger if exact answer matches already exist with `confidence > 0.85`

🧪 Example Output:

```
{
  "tag_id": "payout_policy",
  "question_text": "Is your vendor payout schedule fixed or flexible? How
often do payouts occur?",
  "source": "dynamic_generation",
  "context_used": true,
  "related_segments": ["seg_45"],
  "priority": "core",
  "status": "generated"
}
```

📎 References:

- 📘 *AUDIRA DISCOVERY TAGS DICTIONARY* – prompt templates
- 📘 *AUDIRA AGENT SIMULATION TEST KIT* – used to simulate generated questions
- 📘 *AUDIRA PROMPT CHAIN & LLM LOGIC FLOW* – dynamic phrasing API logic

---

Section 4: Contextual Boosting

🎯 Purpose:

To generate **tailored, human-like follow-up questions** that adapt to each business's data, by injecting context from onboarding answers and document segments when available.

This transforms the tone from *"generic form filler"* to *"insightful advisor."*

When Context Is Used:

1. **Partial Coverage Cases**
   When a tag has a segment or answer with `confidence` between `0.65–0.84`, the system uses that entry to enrich the question.
2. **Contradictory or Ambiguous Inputs**
   If two segments suggest conflicting data, the question explicitly asks for clarification.
3. **Repeated Mentions with Gaps**
   When several segments mention a concept, but leave out a critical detail (e.g. payout frequency, commission %, etc.)

Example Boosted Questions:

Tag	Context Snippet	Generated Follow-up
`payout_policy`	"Vendors are paid every 14 days"	“Is this payout schedule automated or triggered manually?”
`compliance_model`	“We follow GDPR internally”	“Do you also support compliance with other regions like CCPA or PCI?”
`pricing_model`	“Base price is $49/month”	“Are there any variable charges beyond the base price?”

How Context Is Formed:

Source	Type
`onboarding_answers`	Declarative text
`segment_text`	Extracted from uploaded files
`anchor_linkage`	Multi-block summaries (from FSD_03)

The system embeds these and includes them either in:

- The prompt to the LLM
- The final question itself (as a prefix or follow-up)

References:

- 📘 *AUDIRA FILE & DATA UPLOAD SCHEMA* – tracks source of each content block
- 📘 *AUDIRA AGENT ONBOARDING FRAMEWORK* – origin of initial business declarations
- 📘 *AUDIRA PROMPT CHAIN & LLM LOGIC FLOW* – defines LLM injection rules

---

Section 5: Output Format

Unified Question Object Structure

Every dynamically generated follow-up question is saved as a structured object for:

- Display in the onboarding UI
- Feeding into LLM prompt chains
- Feeding into the validator and agent readiness engine

JSON Schema per Question:

```
{
  "question_id": "fq_007",
  "tag_id": "vendor_model",
  "question_text": "How do you typically onboard and verify your vendors?",
  "source": "dynamic_generation",
  "context_used": true,
  "related_segments": ["seg_21", "seg_32"],
  "related_answers": ["q_7"],
  "priority": "core",
  "status": "unanswered",
  "version": 1.0,
  "generation_method": "template+contextual"
}
```

Key Fields:

Field	Description
`question_id`	Unique identifier for tracking
`tag_id`	Links to the business discovery tag it aims to complete
`question_text`	Final natural language question
`source`	`template`, `contextual`, `LLM`, or hybrid
`context_used`	Boolean flag — whether segment or answer was used to shape question
`related_segments`	Segment IDs that influenced the prompt
`related_answers`	Onboarding answers involved in generation
`priority`	core, advanced, or optional (from tag dictionary)
`status`	`unanswered`, `answered`, `skipped`, or `ignored`
`version`	Used for iteration, feedback, retraining logic

Follow-up Metadata:

- Time-to-answer (for analytics)
- Clarification score (if the answer still leaves ambiguity)

- Flagged by validator (FSD_05) as incomplete

📎 References:

- 📘 *AUDIRA DISCOVERY TAGS DICTIONARY* – defines `priority` and template logic
- 📘 *AUDIRA PRE-LAUNCH VALIDATOR SPEC* – uses this object for scoring launch readiness
- 📘 *AUDIRA PROMPT CHAIN & LLM LOGIC FLOW* – uses `question_text` in final prompt injection

---

Section 6: AI Runtime Behavior

🎯 Purpose:

To define **how the generated questions** are delivered to the user — ensuring they feel consultative, not overwhelming. This includes UI timing, batch rules, and fallback strategies.

💮 When Questions Are Shown:

✅ *Final Stage of Onboarding*

After:

- Fixed 10 questions
- 10 dynamic discovery questions
- Document uploads (FSD_01 → FSD_03 outputs complete)

Then the **Dynamic Qualification Step** triggers, where these questions are presented.

💡 *Optional Routing:*

- Via agent interface (if a human onboarding assistant is involved)
- Via client-facing wizard (for self-serve SMBs)
- Via API for white-labeled integrations

🧠 Behavior Logic:

Behavior	Rule

Display Timing	After docs parsed + gap map complete
Display Mode	One-by-one, grouped by theme (recommended)
Max Questions	Default 15 (adjustable)
Skippable?	Yes, but flagged in validator
Editable?	Optional UI flag (admins can modify or rewrite questions)
LLM Prompt Injection	If the customer responds, their answer is linked back to the tag for downstream logic

---

🧪 User Interaction Example:

UI Prompt:

"To finalize your agent's setup, we have a few tailored questions to ensure we fully understand your business."

Follow-up Q:

"How are vendors paid on your platform — automatically or after review?"

Buttons:

- "Answer"
- "Skip"
- "Not Relevant"

Each button triggers storage, tag map update, and validator update.

🔁 Post-Answer Logic:

- Update tag coverage map in real-time
- Feed into validator (FSD_05) to determine launch readiness
- May optionally trigger **second-layer follow-up** if answer still lacks clarity (flagged for admin or AI re-ask)

📎 References:

- 📘 *AUDIRA LAUNCH DEPLOYMENT SHEET* – sets min tag coverage and completion requirements
- 📘 *AUDIRA AGENT ONBOARDING FRAMEWORK* – marks where this stage fits
- 📘 *AUDIRA PROMPT CHAIN & LLM LOGIC FLOW* – consumes answers to improve future prompt clarity

Section 7: Edge Handling

🎯 Purpose:

To gracefully manage edge cases where the system generates:

- **Too few** follow-up questions (under 5)
- **Too many** follow-up questions (over 15)
- Or if the tag confidence map shows conflicting or missing data

This ensures **consistency, flexibility, and user trust** across different SMB profiles and onboarding situations.

📉 Case A: Too Few Questions (< 5)

Condition	System Response
Most tags are already covered with `confidence > 0.85`	Display success message: *"Most of your key areas are well covered!"*
Only optional/low-priority tags remain	Optionally skip dynamic qualification step
Coverage stats flagged as "thin" by Validator	Inject an **admin review flag** or trigger fallback: "Ask 3 general business growth questions"

📈 Case B: Too Many Questions (> 15)

Condition	System Response
Total gaps + new tags > 15	Sort by priority and relevance, show top 15 only
Tags with equal priority	Rank by frequency in segments or confidence weakness
Admin override flag active	Enable **follow-up queue** (show 15 now, more later via dashboard)
Business type = complex (from fixed questions)	Allow override to **20-question mode** (toggle in Launch Config)

🧭 Dynamic Targeting Option:

Audira supports an **auto-adjust mode**:
Instead of static 15 questions, it calculates how many are needed to reach minimum viable tag coverage (per `AUDIRA PRE-LAUNCH VALIDATOR SPEC`).

Example: If validator requires 85% coverage and current state is 60%, generate as many questions as needed to reach the gap.

🚫 Case C: Broken Tag Confidence Map

Symptom	Recovery
Coverage map not generated	Retry FSD_03
Gaps detected but no tags linked	Trigger soft-fail fallback:
"Tell us more about your business model, pricing, and support policies"	
LLM refuses prompt (rate limit, failure)	Use stored templates as fallback source

📎 References:

- 📘 *AUDIRA PRE-LAUNCH VALIDATOR SPEC* – defines coverage thresholds
- 📘 *AUDIRA AGENT SIMULATION TEST KIT* – tests various gap configurations
- 📘 *AUDIRA AGENT ONBOARDING FRAMEWORK* – defines expected dynamic stage behavior

Section 8: Future Enhancements

🚀 Upgrade Roadmap

Feature	Description	Why & When
Multilingual Follow-up Generation	Dynamically generate questions in the SMB's native language (Arabic, French, etc.)	Boosts onboarding efficiency across global markets
Style-Aware Prompting	Adjust tone: formal, friendly, expert, etc., based on industry or brand profile	Ensures brand alignment and improves response rates
Auto-Clustered Question Bundles	Group questions under themes (e.g. Revenue, Vendors, Support) with progressive logic	Makes the experience feel like a conversation, not a form
Adaptive Reasoning Loop	If a response to one question unlocks a new tag, trigger a follow-up chain	Simulates intelligent back-and-forth like a real consultant
Agent-Type Tuning	Tailor questions to the type of agent being deployed (support bot, sales bot, compliance bot)	Hyper-focuses the qualification for faster launch readiness
Time-Based Triggers	If no document has been updated for 30+ days, re-trigger qualification	Keeps agent memory fresh and aligns with business changes

🧰 Open Source Tools (MIT / Permissive)

Tool	Usage
`PromptLayer`, `PromptTools`, `LangChain Prompt Templates`	For dynamic question phrasing with version control
`Haystack QA`, `LlamaIndex`, `Semantic Router`	Tag-document-query alignment logic
`LangGraph`, `LangFlow` (OSS MIT)	Workflow routing if multi-question chaining is enabled
`DeepEval`, `Trulens`	Evaluation of question quality and answer completeness
`Unstructured.io`	Used earlier but helps enrich document block context for better question logic

---

📎 Linked Modules:

- 📘 *AUDIRA DISCOVERY TAGS DICTIONARY* – must evolve alongside to support more tag variations
- 📘 *AUDIRA LAUNCH DEPLOYMENT SHEET* – dynamic Q count logic may evolve based on this
- 📘 *AUDIRA PROMPT CHAIN & LLM LOGIC FLOW* – needed for injecting answers into runtime agent behavior

---