From Paper to Structured JSON: An Agentic AI Workflow for Compliant BMR Digital Transformation

Bhavik Agarwal[*†], **Nidhi Bendre**[*], **Viktoria Rojkova**
MasterControl AI Research
{bagarwal,nbendre,vrojkova}@mastercontrol.com

Abstract

Pharmaceutical manufacturers generate thousands of batch manufacturing records (BMRs) each year. Under FDA 21 CFR Part 211 and EU GMP guidelines, these 100+ page documents mix tables, calculations, images, and handwritten annotations and must be retained for decades (U.S. Food and Drug Administration, 2024b; European Medicines Agency, 2022). Existing options sit between generic document tools that lack pharmaceutical semantics and industry systems that assume already digital, standardized inputs (AWS Partner Network, 2025; LlamaIndex Team, 2024). As a result, most BMRs are still converted and reviewed manually, with effort scaling linearly with volume (Pharmaceutical Technology Editors, 2024).

We present an agentic AI workflow that converts unstructured BMRs into compliant, structured JSON. The system uses hybrid OCR + vision-language document understanding, token-based chunking, and parallel LLM extraction guided by a TypeScript-like schema that encodes the pharmaceutical *Group–Phase–Step* hierarchy and 11 content types (including tables, calculations, numeric/date fields, images, and pass/fail checks). Three validation layers enforce syntactic correctness, structural integrity, and pharmaceutical compliance, and coverage-style metrics expose extraction quality.

On three real-world BMRs (15–66 pages), the system achieves composite confidence scores of 82.08–89.00%, with perfect hierarchy, sequence, and cross-reference preservation, and perfect fidelity for calculations, conditional logic, and units. Processing time drops from several hours of manual quality review per document to minutes or tens of minutes on standard infrastructure (single GPU, up to 8 parallel workers). Remaining challenges include OCR noise on historical documents and cross-chunk context for very long ($>$ 150-page) records. Overall, schema-guided, validated extraction enables practical, human-in-the-loop BMR digitization at scale.

1 Introduction

Batch Manufacturing Records (BMRs) document the complete manufacturing history of a pharmaceutical product, including materials, equipment, process parameters, quality results, deviations, and corrective actions. They are mandated by regulators such as the FDA (21 CFR Part 211) and EMA (EU GMP) and form a core part of the quality system, enabling traceability, recalls, and patient safety safeguards (U.S. Food and Drug Administration, 2024b; European Medicines Agency, 2022; Pharmaceutical Technology Editors, 2024).

In practice, many facilities still rely on paper or scan-based BMRs. Operators transcribe readings by hand, perform calculations without enforced checks, and collect wet signatures. Quality assurance (QA) teams then review 100–150 page records line by line before physical archiving. Retrieving data for investigations or process improvement requires locating and re-reading paper records, often under time pressure (International Society for Pharmaceutical Engineering, 2023; McKinsey & Company, 2025).

Generic LLM-based extractors (e.g., invoice or contract parsers) can produce JSON but lack domain constraints and GMP-specific structures (LlamaIndex Team, 2024; LangChain Team, 2024). Pharmaceutical electronic batch record (EBR) systems, in contrast, assume prospective digital capture and standardized templates and are not designed to ingest heterogeneous historical BMRs (AWS Partner Network, 2025; MasterControl Inc., 2023). The result is a gap: decades of manufacturing knowledge remain locked in un-

[*] Equal contribution.

[†] Current arXiv preprint: *From Paper to Structured JSON: An Agentic Workflow for Compliant BMR Digital Transformation*. arXiv:2601.04368 (2025).

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track), pages 39–47
March 25-27, 2026 ©2026 Association for Computational Linguistics

structured documents (Smith et al., 2024).

This paper describes an agentic AI workflow that bridges this gap. Our contributions are:

- A production-oriented pipeline that transforms unstructured BMR PDFs into structured JSON that preserves the *Group–Phase–Step* hierarchy and key GMP semantics.
- A hybrid document understanding stack combining MarkItDown, OCR, and a vision-language model to handle long, noisy, and handwritten BMRs.
- A schema-guided extraction and validation framework with coverage metrics tailored to pharmaceutical documentation.
- An empirical evaluation on three real BMRs, showing high structural fidelity and useful coverage with large reductions in processing time.

2 Problem

Operational burden. Time-motion studies indicate that each BMR requires roughly three hours of QA review (ISPE Metrics Team, 2023). A mid-size site producing 100 batches per month generates about 1,200 BMRs annually, leading to thousands of QA hours per year and tens of thousands of archived documents over a decade. Operators can spend 30% of their time on documentation rather than value-generating manufacturing work (Johnson and Williams, 2024), delaying batch release and increasing inventory costs.

Error and compliance risk. Manual documentation is a leading cause of deviations and quality incidents (Product Quality Research Institute, 2023; Anderson et al., 2024). Common failure modes include transcription errors, missing signatures, and incorrect or unchecked calculations. Conventional OCR tools can extract text but cannot verify calculations, enforce 21 CFR Part 11–style audit trails, or reliably preserve conditional logic and cross-references (U.S. Food and Drug Administration, 2024a). This creates regulatory risk and costly remediation when problems are detected.

Scalability limits. Manual digitization of historical archives is even less tractable. At conversion rates of 2–3 BMRs per person-day, fully processing 20,000 legacy records would require roughly 27 person-years of effort (Accenture Life Sciences, 2023; Boston Consulting Group, 2024). This cost blocks many organizations from using their own historical process data for yield optimization, deviation trending, or technology transfer.

Technology gap. Current AI-based document extractors are agnostic to GMP constraints and pharmaceutical semantics, while industry EBR systems assume clean, structured inputs. No widely deployed system jointly offers: (i) robust extraction from noisy, heterogeneous BMRs; (ii) strong schema and relationship constraints aligned with GMP practice; and (iii) explicit quality metrics exposing what was reliably captured and what needs human review.

3 System Overview

Our system is designed as a human-in-the-loop workflow that starts from a BMR PDF and ends with structured JSON plus quality signals, ready for downstream use but still subject to human approval.

Input and document understanding. A user uploads a BMR PDF (digital or scanned). The system first applies a hybrid document understanding stack: MarkItDown produces markdown with headings, lists, and tables; a vision-language model extracts text and layout from images and complex regions; and a fallback OCR engine addresses cases where the vision model is not used or fails. The result is a single, enriched markdown representation that includes tables, paragraphs, forms, calculations, and image-derived text.

Hierarchical modeling. BMRs follow a consistent but domain-specific structure: high-level groups (e.g., “Processing”, “Packaging”), phases within groups (e.g., “Material Preparation”, “Blending”), and steps within phases. The system models this *Group–Phase–Step* hierarchy explicitly. Content inside steps is labeled into a compact set of content types, including text, numeric/date fields, choice/pass/fail fields, tables, calculations, timestamps, links, images, and attachments. This structure is encoded in a TypeScript-style schema that becomes the contract for extraction.

Parallel, chunk-based extraction. The markdown is split into token-based chunks that fit within the LLM context window while roughly respecting sentence boundaries and logical section breaks. Chunks are processed in parallel by worker agents that convert their piece of the document into JSON conforming to the schema. Identifiers (group, phase, step IDs) are generated so that cross-chunk relationships can be merged without collision.

Validation, metrics, and user output. After all chunks are processed, a validator merges outputs and applies three layers of checks: (i) JSON and tag syntax; (ii) structural integrity (hierarchy, sequence, ID consistency, cross-references); and (iii) pharmaceutical semantics (calculations, units, conditional logic, field-level completeness). Coverage-style metrics (e.g., crude vs. context-aware word coverage, reference coverage, step accuracy) are combined into a composite confidence score. The user receives the final JSON, a summary of metrics, and flags for low-confidence regions that warrant human review.

4 Technical Architecture

This section summarizes the main technical design decisions that enable robust, production-ready BMR digitization without code-level detail.

4.1 Hybrid document understanding

We evaluated multiple OCR and document understanding models, including IBM Granite Docling, RedNote DOTS OCR, Nanonets OCR, Microsoft TrOCR, and Donut. In practice, a hybrid approach worked best for our use case:

- **MarkItDown** for primary document structure extraction and markdown conversion, preserving headings, lists, and tables.

- **Qwen3-VL-8B-Instruct** (or similar) as the main vision-language model for extracting text and layout from complex regions, tables, stamps, diagrams, and handwritten annotations.

- **Tesseract OCR** with multiple page segmentation configurations as a fallback when the vision model is disabled or fails.

The pipeline first runs MarkItDown, then identifies images and complex regions that benefit from vision-language processing. OCR results are merged back into the markdown so that downstream components see a unified text representation. This hybrid stack improves robustness on low-quality scans and documents with heavy annotation, where pure OCR approaches perform poorly.

4.2 Chunking and parallel processing

BMRs often exceed 100 pages, which would overflow typical LLM context limits if processed as a single sequence. We therefore implement token-based chunking with a greedy sentence-packing strategy and a threshold of roughly 3,000 tokens per chunk. Oversized units (e.g., very large tables) are hard-split when necessary.

Chunks are processed concurrently using a thread pool with up to eight workers, each calling a schema-aware extraction prompt that converts its chunk into JSON. To preserve global structure, workers:

- maintain references to the current group and phase, inferred from headings and section markers; and

- allocate globally unique IDs using a shared range or offset scheme so that merged JSON has consistent `group_id`, `phase_id`, and `step_id` fields.

This design reduces end-to-end runtime from hours to minutes or tens of minutes, while retaining the ability to reason about document-wide relationships.

4.3 Schema-guided extraction and validation

Instead of relying on free-form extraction with post-hoc heuristics, the system uses a TypeScript-like schema to steer the LLM. The schema defines:

- field types (e.g., `"text"`, `"numeric"`, `"date"`, `"choice"`, `"pass_fail"`, `"timestamp"`, `"boolean"`);

- content objects for paragraphs, lists, notes, instructions, data forms, calculations, tables, and images; and

- the `Group`, `Phase`, and `Step` classes, with explicit `id`, `group_id`, and `phase_id` links.

Prompts include the schema and a small set of extraction rules (e.g., “do not nest phases inside groups in the JSON; use IDs instead"), and require valid JSON output only. In our internal experiments, this representation reduced schema violations compared to JSON Schema–style descriptions and made it easier for domain experts to review and adjust types.

After extraction, a validator runs three layers of checks:

1. **Syntactic validation:** JSON parses successfully, arrays and objects are well-formed, and reserved tags are used correctly.

2. **Structural validation:** all phase and step references resolve; sequence ordering matches the source document; cross-references (e.g., "see Table 3") are internally consistent when possible.

3. **Pharmaceutical validation:** calculation expressions, variable names, units, and acceptable ranges are well-formed; pass/fail logic appears consistent; and header-level information (e.g., batch name, SKU, dates) is populated.

Coverage-style metrics estimate how much of the original content was captured and how faithfully. These metrics drive the composite confidence score shown to users and are also used to trigger re-processing or human review of low-confidence sections.

5 Results

We evaluated the system on three representative BMRs from different manufacturing contexts: oral solid encapsulation, contract packaging, and solid-dose tablet manufacturing. The documents range from 15 to 66 pages and include mixed-quality scans, tables, calculations, handwritten annotations, and multi-step procedures.

5.1 Extraction quality

Table 1 summarizes key metrics across the three documents. Coverage metrics capture how much content was extracted; structural metrics capture preservation of the Group–Phase–Step hierarchy and cross-references; and content fidelity metrics capture correctness of calculations, conditional logic, units, and step-level details.

Crude word coverage varies with scan quality and layout, but context-aware coverage—a looser measure of whether the essential meaning of each region is present somewhere in the JSON—remains above 93% for all documents. Pharmaceutical-critical elements (calculations, conditional logic, units) are consistently extracted with 100% fidelity. Structural metrics show perfect preservation of the Group–Phase–Step hierarchy and step ordering.

Step-level accuracy, which requires that all fine-grained fields and notes are correctly captured and associated with the right step, is lower (75–83%) and reflects the main residual error source. Common issues include handwritten annotations overlapping printed text, site-specific abbreviations not seen during development, and multi-page tables with irregular header repetition.

5.2 Processing performance

On a single GPU with up to eight worker threads, processing times fall in the "minutes to tens of minutes" range for 15–66 page BMRs, instead of multiple hours of manual QA review per document. We observe that processing time is influenced more by layout complexity and image density than by page count alone: a shorter but heavily annotated packaging BMR can take longer than a longer but cleaner tablet BMR. Parallel chunk processing scales well up to the tested sizes; extremely long documents ($>$ 150 pages) stress cross-chunk context, as discussed below.

6 Discussion and Future Work

Impact for industry. The workflow directly addresses a common bottleneck in pharmaceutical manufacturing: converting unstructured, paper-based BMRs into digital assets that can be searched, analyzed, and reused. By preserving the domain-specific hierarchy and critical semantics, the system produces outputs that can feed into quality dashboards, deviation trending, yield investigations, and technology transfer, while still allowing QA teams to remain in control through confidence scores and review queues.

Limitations. The main technical limitations relate to: (i) document quality, particularly older scans with heavy handwriting and stamps, where OCR and vision models set an upper bound on achievable fidelity; (ii) cross-chunk reasoning, especially when deviation narratives and corrective actions span many pages and chunk boundaries; and (iii) coverage of local notation and abbrevia-

Table 1: Extraction quality and coverage metrics across three real-world BMRs.

Metric Category	**Encapsulation BMR (%)**	**Sharp Packaging BMR (%)**	**Metformin HCl Tabs BMR (%)**
Coverage Metrics			
Crude Word Coverage	71.33	54.19	67.00
Context-Aware Coverage	94.12	96.00	93.49
Reference Coverage	80.00	100.00	95.00
Structural Integrity			
Hierarchy Preservation	100.00	100.00	100.00
Sequence Preservation	100.00	100.00	100.00
Cross-Reference Integrity	100.00	100.00	100.00
Content Fidelity			
Calculation Fidelity	100.00	100.00	100.00
Conditional Logic	100.00	100.00	100.00
Unit Fidelity	100.00	100.00	100.00
Step Accuracy	82.72	75.09	80.25
Document Characteristics			
Unique Step Types Identified	3	7	7
Composite Confidence Score	**89.00**	**82.08**	**88.77**

tions, which vary by site and product. These factors mostly affect step-level accuracy and reference coverage, rather than high-level hierarchy or calculations.

Ethical and regulatory considerations. The system is explicitly designed as an assistive tool, not an autonomous decision-maker. All outputs are intended to be reviewed and approved by qualified personnel before entering validated GMP systems. Confidence scores are intentionally conservative to reduce automation bias: ambiguous content is flagged for human attention rather than silently accepted. From a data protection perspective, the system supports on-premise deployment, log redaction of personal identifiers, and encrypted storage, but organizations must still implement appropriate access control and retention policies to meet their regulatory obligations.

Future directions. Future work focuses on three areas. First, expanding the pharmaceutical knowledge base used during validation (e.g., GxP documents, guidance on stability, validation, and change control) to catch more subtle compliance issues. Second, improving cross-chunk reasoning through lightweight retrieval or global context summaries, particularly for very long BMRs and deviation chains. Third, extending the system beyond passively processing BMRs toward an "intelligent quality assistant" that can surface recurring failure patterns, suggest process optimizations, and help generate regulatory submission content using the extracted JSON as a foundation.

References

Accenture Life Sciences. 2023. Digital transformation in life sciences: The document challenge. Industry Report.

L. Anderson and 1 others. 2024. Documentation quality and its impact on pharmaceutical manufacturing outcomes. *Journal of Pharmaceutical Sciences*, 113:1567–1579.

AWS Partner Network. 2025. Digitalizing batch records in pharmaceutical production with Aizon. AWS Partner Network Blog.

Boston Consulting Group. 2024. Economic analysis of pharmaceutical manufacturing scale-up. Industry analysis, BCG.

European Medicines Agency. 2022. EU Guidelines for Good Manufacturing Practice for Medicinal Products. EudraLex Volume 4.

International Society for Pharmaceutical Engineering. 2023. Pharmaceutical manufacturing digitization: Current state and future trends. Industry report, ISPE.

ISPE Metrics Team. 2023. Manufacturing metrics and KPIs in pharmaceutical production. *Pharmaceutical Engineering*, 43(4).

R. Johnson and K. Williams. 2024. Lean manufacturing applications in pharmaceutical production. *Journal of Pharmaceutical Innovation*, 19:234–251.

LangChain Team. 2024. LangChain: Building applications with LLMs. Technical Documentation.

LlamaIndex Team. 2024. LlamaExtract: Document extraction with LLMs. Software Documentation.

MasterControl Inc. 2023. Electronic batch record systems: Implementation and benefits. White Paper.

McKinsey & Company. 2025. Gen AI: A game changer for biopharma operations.

Pharmaceutical Technology Editors. 2024. Batch manufacturing records: Best practices for compliance and efficiency. *Pharmaceutical Technology*, 48(3).

Product Quality Research Institute. 2023. Analysis of manufacturing deviations in pharmaceutical production: A multi-site study. *PDA Journal of Pharmaceutical Science and Technology*, 77(5).

J. Smith and 1 others. 2024. Artificial intelligence in pharmaceutical manufacturing: Progress and challenges. *Nature Reviews Drug Discovery*, 23:45–62.

U.S. Food and Drug Administration. 2024a. 21 CFR Part 11 - Electronic Records; Electronic Signatures. Guidance document, FDA.

U.S. Food and Drug Administration. 2024b. 21 CFR Part 211 - Current Good Manufacturing Practice for Finished Pharmaceuticals. Electronic Code of Federal Regulations.

Appendix

A Complete TypeScript Schema Template

Listing 1: Full TypeScript Schema for BMR Extraction

```
type FieldType = "text" | "numeric" | "
    date" | "choice" |
                 "pass_fail" | "
    timestamp" | "boolean";

class Field {
    type: FieldType[];
    value: any;
    constructor(type: FieldType[], value
    : any) {
        this.type = type;
        this.value = value;
    }
}

class Header {
    completion_date: Field;
    expiry_date: Field;
    name: Field;
    quantity: Field;
    sku: Field;
    start_date: Field;

    constructor() {
        this.completion_date = new Field
    (
            ["date"],
            "The date the batch process
    was completed"
        );
        this.expiry_date = new Field(
            ["date"],
            "Expiration date of the
    final product batch"
        );
        this.name = new Field(
            ["text"],
            "Name of the batch record
    template"
        );
        this.quantity = new Field(
            ["numeric"],
            "The quantity or yield of
    the final product"
        );
        this.sku = new Field(
            ["text"],
            "Stock Keeping Unit
    identifier"
        );
        this.start_date = new Field(
            ["date"],
            "Date when the batch process
     started"
        );
    }
}

class Content {
    type: "paragraph" | "bullet_list" |
    "numbered_list" |
          "note" | "warning" | "
    instruction" | "data_form" |
          "calculation" | "table" | "
    image";
    text: string;
    items?: string[];
    fields?: {
        label: string;
        value: string | null;
        unit?: string;
        limits?: string;
        notes?: string;
    }[];
    calculation?: {
        formula: string;
        variables: {
            name: string;
            description: string;
            value?: any;
            unit?: string;
        }[];
        result?: {
            value: any;
            unit?: string;
        };
        notes?: string;
    };
    headers?: string[];
    rows?: any[][];
}

class Step {
    id: string;
    phase_id: string;
    group_id: string;
    step_name: Field;
    step_type: Field;
    content: Content[];
}

class Phase {
    id: string;
    group_id: string;
    phase_name: Field;
}

class Group {
    id: string;
    group_name: Field;
}
```

B Extraction Prompts

B.1 First Chunk Prompt

Listing 2: Prompt Template for Initial Chunk

```
Please convert the following
    manufacturing batch record
(chunk {chunk_number} of {total_chunks})
     into a structured
JSON format according to the provided
    template.

Input:
- Manufacturing Batch Record: {mbr}
- Template Structure: {template}

Requirements:
1. Generate a complete, valid JSON that
    strictly follows
   proper JSON syntax
2. Your JSON MUST contain separate top-
    level arrays for
   groups, phases, and steps:
   {
       "header": {general information
    about the document},
       "groups": [array of Group objects
    ],
       "phases": [array of Phase objects
    ],
       "steps": [array of Step objects]
   }
3. Do NOT nest phases inside groups or
    steps inside phases
4. CRITICAL JSON SYNTAX REQUIREMENTS:
   a) Use only valid JSON syntax - NO
    JavaScript functions
   b) Do NOT use TypeScript class
    initialization syntax
   c) For empty arrays, use [] not Array
    ()
   d) Ensure all table rows have the
    same number of columns
5. Each object must include ALL fields
    defined in its class
6. Include ALL relevant information from
     the batch record
7. IMPORTANT: When encountering text
    from images (indicated
   by "[Image Text: ...]"), create
    content objects with
   type "image" and place the extracted
    text in "text" field

Wrap your response in <json></json> tags
     as follows:
<json>
{
    "header": {...},
    "groups": [...],
    "phases": [...],
    "steps": [...]
}
</json>

Ensure your JSON is fully parsable - no
    syntax errors,
unclosed brackets, or trailing commas.
```

C Example Input and Output

C.1 Sample Input Markdown (Partial)

Listing 3: Example BMR Markdown Input

```
# BATCH MANUFACTURING RECORD
**Product:** Acetaminophen Tablets 500mg
**Batch Number:** AT-2024-0156
**Manufacturing Date:** 2024-03-15

## EQUIPMENT REQUIRED
| Equipment | ID Number | Calibration
    Due |
|-----------|-----------|-----------|
| V-Blender | VB-105 | 2024-04-20 |
| Tablet Press | TP-203 | 2024-05-15 |
| Metal Detector | MD-089 | 2024-03-30 |

## PROCESSING INSTRUCTIONS

### Phase 1: Material Preparation
**Step 1:** Weigh acetaminophen powder
- Target weight: 50.0 kg +/- 0.5 kg
- Actual weight: ________ kg
- Performed by: ________ Date: ________

**Step 2:** Screen acetaminophen through
     20 mesh
- Pass all material through screen
- Record any retained material: ________
     g
- [Image Text: Screening setup diagram
    showing
  20 mesh screen positioned above
    collection bin]

### Phase 2: Blending
**Step 3:** Load materials into V-
    blender
- Add screened acetaminophen
- Add microcrystalline cellulose: 5.0 kg
- Blending time: 15 minutes
- Blender speed: 12 rpm

**Calculation:** Theoretical Yield
Formula: (Acetaminophen + Excipients) x
    0.98
Variables:
- Acetaminophen weight: 50.0 kg
- Total excipients: 7.5 kg
Expected yield: 56.35 kg
Acceptable range: 95.0% - 103.0%
```

C.2 Expected JSON Output Structure

Listing 4: Example Structured JSON Output

```
{
    "header": {
        "completion_date": {"type": ["
    date"], "value": null},
        "expiry_date": {"type": ["date
    "], "value": null},
        "name": {"type": ["text"],
                 "value": "Acetaminophen
     Tablets 500mg"},
        "quantity": {"type": ["numeric
    "], "value": null},
        "sku": {"type": ["text"], "value
    ": "AT-2024-0156"},
        "start_date": {"type": ["date"],
     "value": "2024-03-15"}
    },
    "groups": [
        {
            "id": "group-1",
            "group_name": {"type": ["
    text"], "value": "Processing"}
        }
    ],
    "phases": [
        {
            "id": "phase-1",
            "group_id": "group-1",
            "phase_name": {"type": ["
    text"],
                          "value": "
    Material Preparation"}
        },
        {
            "id": "phase-2",
            "group_id": "group-1",
            "phase_name": {"type": ["
    text"], "value": "Blending"}
        }
    ],
    "steps": [
        {
            "id": "step-1",
            "phase_id": "phase-1",
            "group_id": "group-1",
            "step_name": {"type": ["text
    "],
                         "value": "Weigh
     acetaminophen powder"},
            "step_type": {"type": ["
    numeric"], "value": null},
            "content": [
                {
                    "type": "data_form",
                    "text": "Weight
    measurement form",
                    "fields": [
                        {
                            "label": "
    Target weight",
                            "value":
    "50.0",
                            "unit": "kg
    ",
                            "limits":
    "+/- 0.5 kg"
                        },
                        {
                            "label": "
    Actual weight",
                            "value":
    null,
                            "unit": "kg"
                        }
                    ]
                }
            ]
        },
        {
            "id": "step-2",
            "phase_id": "phase-1",
            "group_id": "group-1",
            "step_name": {"type": ["text
    "],
                         "value": "
    Screen acetaminophen through 20 mesh
    "},
            "step_type": {"type": ["text
    "], "value": null},
            "content": [
                {
                    "type": "instruction
    ",
                    "text": "Pass all
    material through screen"
                },
                {
                    "type": "image",
                    "text": "Screening
    setup diagram showing 20 mesh \
screen positioned above collection bin"
                }
            ]
        },
        {
            "id": "step-3",
            "phase_id": "phase-2",
            "group_id": "group-1",
            "step_name": {"type": ["text
    "],
                         "value": "Load
    materials into V-blender"},
            "step_type": {"type": ["text
    "], "value": null},
            "content": [
                {
                    "type": "bullet_list
    ",
                    "text": "Materials
    to add",
                    "items": [
                        "Add screened
    acetaminophen",
                        "Add
    microcrystalline cellulose: 5.0 kg"
                    ]
                },
                {
                    "type": "calculation
    ",
                    "text": "Theoretical
     Yield Calculation",
                    "calculation": {
                        "formula": "(
    Acetaminophen + Excipients) x 0.98",
                        "variables": [
                            {
                                "name":
    "Acetaminophen",
                                "
    description": "Weight of active
    ingredient",
                                "value":
     50.0,
                                "unit":
    "kg"
                            },
                            {
                                "name":
    "Excipients",
                                "
    description": "Total excipient
    weight",
                                "value":
     7.5,
                                "unit":
    "kg"
                            }
                        ],
                        "result": {
                            "value":
    56.35,
                            "unit": "kg"
                        },
                        "notes": "
    Acceptable range: 95.0% - 103.0%"
                    }
                }
            ]
        }
    ]
}
```