

can you write a detailed functional requirements document

Absolutely! Below is a **Functional Requirements Document (FRD)** that covers both `callable_bond_metrics.py` and `non_callable_bond_metrics.py`. This FRD is structured for clarity and maintainability, suitable for submission to project management or dev teams.

Functional Requirements Document (FRD)

1. Overview

This document specifies the functional requirements for a bond analytics tool capable of calculating key risk and return metrics for both callable and non-callable fixed rate bonds using the QuantLib library. The tool should accurately compute and report:

- Yield to Maturity (YTM)
- Yield to Worst (YTW, for callable bonds)
- Modified Duration
- Macaulay Duration (for non-callable bonds)
- Convexity (standard and effective)
- Effective Duration and Convexity (for non-callable bonds)

The tool will be implemented as Python modules and scripts, allowing for flexible extension and integration.

2. Scope

- Calculate and print metrics for:
 - Callable fixed rate bonds
 - Non-callable fixed rate bonds
- Enable pricing using user-specified yield curves
- Output all results in a human-readable format for verification and further analysis

3. Functional Requirements

3.1. Input Parameters

3.1.1. General Parameters (Both Bond Types)

Parameter	Type	Description	Example
Evaluation date	Date	QuantLib Date object for pricing/evaluation	2025-04-21
Calendar	Enum	Calendar for business day adjustments	UnitedStates
Business convention	Enum	E.g., Following, ModifiedFollowing	Following
Day count	Enum	Day count convention for calculations	ActualActual
Settlement days	Integer	Number of days to settle	2
Face amount	Float	Par value of the bond	1000.0
Coupon rate	Float	Annual coupon rate (decimal, not %)	0.05
Frequency	Enum	Coupon frequency	Semiannual
Issue date	Date	Date bond is issued	2020-04-21
Maturity date	Date	Date bond matures	2030-04-21
Flat market rate	Float	Flat rate used for the discount curve	0.04

3.1.2. Callable Bond-Specific Parameters

- **Call Schedule:** List of call dates (and associated call prices).
 - Example: Callable on April 21 of each year from 2026 to 2029 at 100.0 (clean).

3.2. Metrics & Calculations

3.2.1. Non-Callable Bond

- **Yield to Maturity (YTM):** Calculate via QuantLib's `bondYield` method.
- **Modified Duration:** Use `BondFunctions.duration` with `Duration.Modified`.
- **Macaulay Duration:** Use `BondFunctions.duration` with `Duration.Macaulay`.
- **Convexity:** Use `BondFunctions.convexity`.
- **Effective Duration & Convexity:** Compute via price-sensitivity to small parallel shifts in yield curve.

Detailed Steps

1. Initialization:

All bond and market parameters are used to initialize a `FixedRateBond`. The discounting engine is set to use the flat market rate.

2. Yield Calculation:

Use `bond.bondYield(...)` to calculate YTM.

3. Risk Measures:

Leverage QuantLib's convenience functions for duration and convexity.

4. Effective Measures:

Recalculate the price for small shifts ($\pm 1\text{bp}$) in the discount rate and calculate effective duration/convexity as:

- $\text{Eff Duration} \approx (\text{Down_Price} - \text{Up_Price}) / (2 * \text{Base_Price} * \text{shift_size})$
- $\text{Eff Convexity} \approx (\text{Up_Price} + \text{Down_Price} - 2 * \text{Base_Price}) / (\text{Base_Price} * \text{shift_size}^2)$

5. Output:

Print all metrics (YTM, durations, convexity) with appropriate labels.

3.2.2. Callable Bond

- **Yield to Worst (YTW):**

For each call date, calculate yield to call; identify the lowest of these ("worst" yield).

- **Modified Duration to Worst:**

Compute duration for each yield-to-call and report duration corresponding to YTW.

- **Convexity to Worst:**

Similarly, report convexity at the YTW.

- **Output:**

Print all "To Worst" metrics with clear labels.

Detailed Steps

1. Initialization:

Instantiate `CallableFixedRateBond` with callability schedule and schedule of calls.

2. Yield Analysis:

For each call date:

- Calculate the yield needed to discount cash flows to the clean price, assuming call at that date.
- Skip call dates where calculation fails.

3. Find Worst Metrics:

Select minimum yield as YTW and associated duration and convexity.

4. Output:

Print Yield to Worst, Modified Duration to Worst, Convexity to Worst.

3.3. User Interface

- Scripts run as standalone programs (`python callable_bond_metrics.py` or `non_callable_bond_metrics.py`)
- Input parameters are hardcoded in script; future versions may allow CLI or config file input.
- Output is printed to standard output, with decimals rounded to 6 places.

3.4. Error Handling & Edge Cases

- If YTW cannot be calculated for any call date (e.g., price too high to reach), indicate "N/A".
- All exceptions during metric calculations are caught, and computation proceeds to the next call date.

3.5. Extensibility Considerations

- Users should be able to easily modify bond parameters at the top of each script.
- Functions should be decoupled for testability and future packaging as a module.

4. Non-Functional Requirements

- Must use QuantLib-Python as backend.
- Scripts must run with Python 3.x.
- Software should execute in <2 seconds for a single bond on commodity hardware.
- Output must be in English with internationalization support considered for future versions.

5. Deliverables

- `callable_bond_metrics.py`: Callable bond analytics script
- `non_callable_bond_metrics.py`: Non-callable bond analytics script
- This Functional Requirements Document

6. Out-of-Scope

- Bonds with floating coupons
- Embedded options other than calls (e.g., puts, convertibles)
- GUI/REST API

7. Acceptance Criteria

- For a set of test parameters, script outputs match hand-checked calculations or major vendor systems (e.g., Bloomberg, Refinitiv).
- All labels and output metrics are present and understandable.
- Scripts handle missing yield/convergence situations gracefully.

End of Document

✱✱