Meta-Prompting Studio for Corporate Valuation with LLMs

ApexCo workshop companion | live exercises for compiling, testing, and revising valuation prompts

This studio handout is the second companion to **DCF Models with LLMs I**, **DCF Models with LLMs II**, and the handout **Meta-Prompting for Corporate Valuation with LLMs**. The first handout explained the method. This one makes students practice it. Working with ApexCo rather than the final-project company, students will compile, test, debug, and log a small set of stage prompts that map directly onto the DCF workflow.

Studio promise

A strong team should leave the workshop with four reusable artifacts: **(i)** an architecture prompt package, **(ii)** a fact-bank prompt package, **(iii)** a forecast-bridge prompt package, and **(iv)** a valuation-control prompt package. The goal is not to write a giant prompt. The goal is to produce stage-specific prompts that generate spreadsheet-ready artifacts.

1. Stage brief lock the object, packet, artifact, and taboos

2. Compile use a meta-prompt to generate the stage prompt package

3. Test run the generated prompt in a fresh work thread

4. Revise fix weak retrieval, weak schema, or weak controls

5. Log record what changed and why it mattered

1. How to use this handout

Recommended class format

This handout works best in pairs or trios. One student acts as the **pilot** and writes the prompt. One acts as the **auditor** and checks source boundaries, schema discipline, and method consistency. A third student, if present, acts as the **recorder** and maintains the mini Prompt_Log. The exercise works best when students treat prompt design as analyst work rather than casual chat.

Workshop block	Suggested time	What teams produce	What the instructor should listen for
Warm-up	8–10 min	A repaired weak prompt and one method gate	Are students separating the valuation object from the task request?
Exercise 1	15–18 min	Architecture prompt package + first live output	Are prompts requiring packet-supported language and a model-ready table?
Exercise 2	15–18 min	Fact-bank prompt package + revised schema	Are prompts separating FACT, INFERENCE, and OPEN QUESTION?
Exercise 3	15–18 min	Forecast-bridge prompt package + revision notes	Are prompts translating evidence into assumptions without outsourcing judgment?
Exercise 4	15–18 min	Valuation-control prompt package + red-team add-on	Are prompts surfacing consistency checks across WACC, continuing value, bridge, and multiples?
Debrief	8–12 min	One-page log and transfer note	Can students explain why their second prompt was better than their first?

Live-work rules

1. **Freeze the packet slice before you prompt.** Do not ask for "anything relevant." Name the packet and name the stage.
2. **Use separate threads for compile and work.** One thread creates the prompt package; a fresh thread runs it.
3. **Demand source labels.** Exact quote, table label, period, or packet anchor should appear in every useful retrieval output.
4. **Separate fact from interpretation.** If the model mixes them, revise the schema before moving on.
5. **Revise at least once.** The workshop is about prompt improvement, not first-try perfection.
6. **Log one failure.** A good analyst notices where the prompt failed and fixes the design.

2. ApexCo workshop packet | quick reference

Why the packet is intentionally compact

These exercises are about prompt design, not packet volume. The workshop therefore uses a compressed ApexCo packet that still preserves the full DCF workflow: business architecture, historical drivers, forecast bridge, continuing value, WACC, EV-to-equity bridge, and cross-checks. Students should treat this as a training set for the method, not as a substitute for full-source analysis.

Packet anchor	What it contains	Why it matters in the studio
10-K excerpts 1–7	Segment definitions, installed base, contract share, controls attach, cost pools, capex cycle, backlog-normalization risk	Exercise 1 and Exercise 2
FY2024 earnings release excerpts 1–4	2025 guide, segment framing, margin framing, capex framing	Exercise 2 and Exercise 3
Q4 / FY2024 call excerpts	Demand mix, price-cost, automation timing, installed-base monetization, medium-term economics	Exercise 2 and Exercise 3
Q1 2025 10-Q excerpts 1–5	Segment performance, backlog normalization, service resilience, working-capital timing, peak capex update	Exercise 2 and Exercise 3
Historical driver table	Revenue, EBIT, NOPLAT, invested capital, ROIC, capex, D&A, NOWC	Exercise 3
Illustrative market-input sheet	Risk-free rate, peer beta sheet, borrowing spread, target leverage	Exercise 4
Illustrative bridge inputs	Debt, lease liabilities, other fixed claims, cash, operating cash, diluted shares	Exercise 4

2.1 Selected operating history

Core operating history	2022A	2023A	2024A	Why students should care
Revenue	430	465	500	Gives the base-year size and top-line trend
EBIT	45.6	51.6	59.0	Shows improving operating quality before the forecast
NOPLAT	34.8	39.5	45.0	Starting point for ROIC and FCFF
Average invested capital	265	291	315	Links income to the capital base
ROIC	13.1%	13.6%	14.3%	Shows economic returns already achieved
Capex	34	39	45	Reveals the current investment cycle
D&A	25	26	28	Helps separate maintenance from growth investment
Net operating working capital	58	63	70	Shows cash absorbed by operations

Segment and KPI history	2022A	2023A	2024A	Why students should care
Equipment revenue	310	330	350	Largest revenue engine
Service & Parts revenue	80	89	100	Recurring engine
Controls revenue	40	46	50	Fastest-growing and highest-quality engine
Equipment segment margin	8.0%	8.6%	9.0%	Lower-margin hardware economics
Service segment margin	18.0%	18.5%	19.0%	Better recurring economics
Controls segment margin	29.0%	29.5%	30.0%	Highest incremental profitability
Installed base (000 systems)	84	90	95	Feeds Service and Controls opportunity
Service revenue under contract	56%	58%	61%	Visibility and recurrence
Service contract renewal rate	89%	90%	91%	Retention quality
Controls attach rate	34%	37%	40%	Software penetration and mix uplift
Equipment backlog at year-end	118	123	110	Near-term demand and normalization signal

2.2 Selected packet excerpts students can quote directly

Packet anchor	Condensed excerpt	Why it helps in prompts
10-K excerpt 2	About 65% of Equipment revenue is tied to replacement demand; the remainder is tied to new construction and retrofit projects.	Growth architecture and cyclicality

Packet anchor	Condensed excerpt	Why it helps in prompts
10-K excerpt 3	Installed base is about 95,000 active systems; more than 60% of Service revenue is under recurring contract arrangements; renewals remained above 90%.	Service durability and margin quality
10-K excerpt 4	Controls software was attached to about 40% of new equipment placements in 2024 versus 34% in 2022; retrofit opportunity remains sizable.	Controls growth and mix uplift
10-K excerpt 6	Capex rose to 45 in 2024 due to Ohio automation and heat-pump capacity expansion; the strategic capex cycle is expected to peak in 2025 before moderating.	Reinvestment and FCFF timing
10-K excerpt 7	2025 reported Equipment growth could be affected by backlog conversion and distributor inventory normalization.	Near-term Equipment risk
Earnings release excerpt 1	2025 guide calls for 5% to 7% revenue growth, flat to up modest operating margin, and temporarily constrained free cash flow.	Company guide check, not direct forecast
Earnings release excerpt 3	Automation savings and service-mix gains help margin, but temporary under-absorption and launch costs partly offset the benefit in 2025.	Margin bridge and fade logic
Earnings release excerpt 4	2025 is the peak year of strategic capital spending; capex should move closer to depreciation plus modest growth investment beginning in 2026.	Capex normalization and continuing value
Q4 / FY2024 call, CFO	Future margin improvement should come more from mix and automation than from pricing alone.	What should and should not drive the margin thesis
Q4 / FY2024 call, COO	The automation program becomes most visible in the back half of 2025 and in 2026; the investment enters the books before the full productivity and revenue benefit.	Why turnover and FCFF lag
Q1 2025 10-Q excerpt 1	Q1 revenue rose 5%; Equipment +2%, Service +7%, Controls +11%; launch costs and under-absorption partially offset mix benefits.	Near-term segment confirmation
Q1 2025 10-Q excerpt 4	Inventory rose modestly due to heat-pump launch staging and new automation-cell commissioning; receivables rose with volume.	Working capital and reinvestment timing

3. The reusable compiler template

A good prompt package has five parts

A prompt package is not just one user message. In this workshop, a complete package has **(i)** a short method gate, **(ii)** a generated system prompt, **(iii)** a starter user message, **(iv)** an output schema, and **(v)** a verification checklist. Students should judge prompt quality by the usefulness of the resulting artifact, not by how clever the wording sounds.

Reusable compiler skeleton

```
Design the best reusable prompt package for an LLM research assistant supporting a DCF valuation
stage.
```

```
Stage name: [STAGE NAME].
Valuation object: [ENTERPRISE DCF / FCFF / BRIDGE / MULTIPLES CHECK].
Company: ApexCo.
Allowed packet: [NAME THE PACKET SLICE].
Stage objective: [ONE SENTENCE].
Required artifact: [TABLE / CHECKLIST / MEMO / BRIDGE MAP].
Hard constraints: use packet-supported language only; require source labels; separate FACT from
INFERENCE; do not invent numbers; do not move to later valuation stages.
Return: (1) a system prompt, (2) a starter user message template, (3) a strict output schema, (4)
three failure-mode warnings, and (5) a verification checklist for the analyst.
```

Four questions before you run the compiler

Before pressing enter, ask: **What exact artifact do I want? What packet is allowed? What must the model not do? What would make the output spreadsheet-ready?** If any of those answers is vague, the compiler prompt is still too weak.

4. Exercise 0 | Warm-up repair

Weak one-shot request

```
Read the ApexCo packet and tell me what matters for the valuation.
```

Your task

Repair that request in two moves only:

1. Write a **method gate** of no more than four lines.
2. Write a **compiler prompt** that will generate a better stage prompt for the business-architecture step.

Do not run the prompt yet. This warm-up is about stage definition, not execution.

Warm-up target	What strong work usually includes
Method gate	FCFF / enterprise-value lens; ApexCo packet; architecture step only; output is a table, not a forecast
Compiler prompt	Named packet slice; explicit output schema; packet-supported language; source labels; no numerical forecast; no generic strategy summary

5. Exercise 1 | Compile an architecture prompt package

Stage objective

Build a prompt package that produces a clean **segment engine table**, a compact **business blueprint**, and a short **KPI ledger**. The point is to create a retrieval scaffold that later prompts can use.

Stage brief field	Studio setting
Allowed packet	10-K excerpts 1–5; selected segment history
Required artifact	One table with Engine, Who pays / what is sold, Economic role, Why the model separates it; plus a short blueprint and KPI ledger
Must not do	Must not forecast growth or margins; must not collapse Service and Controls into one bucket; must not write a generic company-overview paragraph
Why this stage matters	Every later fact-bank prompt depends on this architecture and KPI vocabulary

Student deliverable

Use the reusable compiler skeleton to generate the architecture prompt package. Then open a fresh work thread, run the generated prompt, and save the first output. Revise the package once if the model does any of the following: **(i)** merges economic engines, **(ii)** writes strategy language instead of operating language, or **(iii)** fails to create a model-ready table.

What strong teams usually change after run 1

Most first-pass prompts are still too broad. Strong teams usually tighten the **output schema**, narrow the **packet slice**, and add a harder instruction to use **commercial language rather than generic corporate language**. This usually improves table quality more than adding extra adjectives to the prompt.

Debrief question	What to look for
Did the prompt create a retrieval scaffold?	The result should name the engines, monetization logic, and KPIs that later prompts can reuse
Did the prompt oversummarize?	A generic company summary is less useful than an analyst-ready architecture table
Did the prompt stay inside the stage boundary?	Growth claims or margin opinions belong later, not here

6. Exercise 2 | Compile a fact-bank prompt package

Stage objective

Build a prompt package that retrieves a **Growth / Margin / Reinvestment Fact Bank** tied to ApexCo's actual engines and KPIs. The output should help an analyst fill a facts bank or notes column in a valuation workbook.

Stage brief field	Studio setting
Allowed packet	10-K excerpts 2–7; earnings release excerpts 1–4; call excerpts; Q1 2025 10-Q excerpts 1–5; selected history
Required artifact	A table with Topic, Exact fact or quotation, Packet anchor, Structural vs near-term, KPI / line item touched, Possible forecast implication, Open question
Must not do	Must not assign final forecast numbers; must not mix unsupported inference into the fact column; must not use unlabeled prose
Why this stage matters	The forecast should later be traceable back to a packet-supported evidence bank

Student deliverable

Compile the fact-bank prompt package and run it in a fresh thread. If the output mixes facts and interpretation, revise the schema rather than arguing with the prose after the fact. Strong teams often force the model to create three explicit tags: **FACT**, **INFERENCE**, and **OPEN QUESTION**. If you do that, the output usually becomes much easier to audit.

Typical failure modes

- **Generic growth summary.** The model writes about macro demand instead of ApexCo's actual drivers.
- **No period discipline.** 2024 structural facts and Q1 2025 updates get blended without labels.
- **No causal link.** A quote is returned, but the model does not say which forecast line it could influence.
- **Premature forecasting.** The model jumps from evidence to exact numbers even when the stage brief forbids it.

What strong work looks like here

A high-quality fact bank does not tell the analyst what the forecast *must* be. It does something better: it makes the later judgment call easier because the evidence is already organized by driver, time horizon, and model line. In other words, the fact bank should lower cognitive load without pretending to replace judgment.

7. Exercise 3 | Compile a forecast-bridge prompt package

Stage objective

Build a prompt package that converts the fact bank and historical drivers into a **forecast bridge**. The bridge should translate evidence into assumption-ready reasoning for revenue, margins, reinvestment, ROIC, FCFF timing, and continuing value. The prompt must help the analyst reason; it must not silently choose the final numbers.

Stage brief field	Studio setting
Allowed packet	Fact-bank output from Exercise 2; historical driver table; segment and KPI history; management-guidance excerpts
Required artifact	A table with Driver, Historical anchor, Evidence summary, Direction, Temporary vs structural, Candidate range or shape, Counterevidence, What the analyst still must decide
Must not do	Must not produce a fully valued DCF; must not hide the remaining analyst judgment; must not ignore reinvestment when discussing growth
Why this stage matters	This is where facts become a model story: growth, margins, net investment, ROIC, and continuing value must move together

Student deliverable

Compile and run the forecast-bridge prompt package. Then revise it once with one goal only: force better treatment of **temporary versus structural** effects. Strong teams usually discover that one of the hardest prompt-design problems in valuation is getting the model to distinguish a one-year drag from a durable economic feature.

Minimum issues the prompt should surface

- Equipment growth is affected by **replacement demand** and **backlog normalization** at the same time.
- Margin improvement should come more from **mix and automation** than from **price-cost** alone.
- 2025 appears to be the **peak capex year**, so FCFF timing should lag operating improvement.
- Continuing value should reflect a mature ApexCo in which the capex cycle has normalized and returns remain above the cost of capital only if the business mix supports it.

Good analyst behavior at this stage

This is the right stage to ask the model for **candidate ranges**, **fade shapes**, **counterarguments**, and **falsifiers**. It is the wrong stage to ask the model to "pick the numbers." A professional workflow treats the model as a structured debate partner, not as the final owner of assumptions.

8. Exercise 4 | Compile a valuation-control prompt package

Stage objective

Build a prompt package that performs a **valuation control pass**. The package should retrieve and organize WACC evidence, classify EV-to-equity bridge items, and run a consistency audit on continuing value, bridge logic, and cross-check multiples.

Stage brief field	Studio setting
Allowed packet	Illustrative market-input sheet; debt / lease / cash / shares inputs; continuing-value assumptions from the carried-forward ApexCo case
Required artifact	Three compact tables: WACC evidence map, bridge-classification map, and consistency-audit table
Must not do	Must not pretend to know market inputs that are not in the packet; must not jump directly to a target price; must not use P / E to check enterprise value
Why this stage matters	Many bad valuations fail because of consistency mistakes rather than forecasting mistakes

8.1 Mini input sheet for the control pass

Input	Illustrative value	Why it is here
10-year Treasury yield	4.25%	Risk-free-rate anchor
Peer-informed ApexCo beta	0.95	Business-risk input for cost of equity
Equity risk premium	5.50%	CAPM input
Pre-tax cost of debt	5.75%	Current marginal borrowing-cost anchor
Target capital structure	80% equity / 20% debt	Market-value WACC weights
Tax rate	25.0%	Convert debt cost to after-tax basis
Gross debt	120.0	Debt-like claim in bridge
Lease liabilities	20.0	Debt-like claim in bridge if treated as financing-like
Other fixed claims	5.0	Small bridge deduction
Cash and liquid investments	55.0	Starting cash balance
Required operating cash	25.0	Cash needed for operations
Diluted shares outstanding	72.6	Per-share denominator
Base continuing-value assumptions	3.0% growth; 12.5% terminal RONIC	Continuing-value control anchor

Student deliverable

Compile the control prompt package and run it. Then add one **red-team extension** that asks the model to look specifically for double counting or omission across **leases**, **cash**, **diluted shares**, and the **multiple used for reasonableness checks**. A strong red-team extension is often more valuable than a long first-pass prompt.

Consistency traps the prompt should catch

- Using book-value capital structure instead of target market-value weights in WACC.
- Treating lease liabilities one way in the operating story and another way in the bridge.
- Adding back all cash instead of only excess cash.
- Using an undiluted share count in the per-share value.
- Comparing an FCFF DCF to an **equity** multiple rather than an **enterprise** multiple.

9. Optional stretch | Compile a memo-writing prompt under hard constraints

Stretch objective

Design a prompt package that writes a **150–180 word valuation summary** from a completed model-output table only. The prompt should force the model to name the value range, the central operating thesis, one key risk, and one reason the multiple cross-check is not a substitute for the DCF.

Hard constraints for the stretch prompt

Use only the provided output table. Do not invent company facts, quotes, or catalysts. Do not restate every model line. Write in analyst prose, not marketing prose. End with one sentence explaining what evidence would most likely change the valuation view.

Why this stretch matters

Many students assume that writing is the easy part. In practice, summary writing is where unsupported claims often enter. A good memo-writing prompt is narrow, evidence-bound, and numerically disciplined.

10. Debrief | what students should learn from the studio

The three big takeaways

1. **Prompt quality is workflow quality.** Better prompts produce better artifacts because they lock the stage, the packet, and the output schema.
2. **Most improvements come from narrower scope and better schema.** They do not come from more heroic wording.
3. **Revision is part of the method.** If students cannot explain why prompt version 2 was better than version 1, they have not yet fully learned the process.

Transfer note

When moving from ApexCo to the final project, do not copy the language of these exercises mechanically. Instead, carry over the **structure**: stage brief, compiler, work thread, revision pass, and audit log. Replace the company vocabulary, packet anchors, peer logic, and bridge details with those of the target company.

A. Appendix A | One-page stage-brief template

Field	What to fill in
Stage name	Architecture / Fact Bank / Forecast Bridge / WACC / Bridge / Control / Memo
Valuation object	FCFF DCF, enterprise value, equity bridge, or multiples cross-check
Target artifact	Table, checklist, memo, map, or audit sheet
Allowed packet	Exact documents, tables, or workbook tabs the model is allowed to use
Required columns or sections	The schema that makes the output spreadsheet-ready
Must not do	The stage boundary and taboos
Verification check	What the human analyst will inspect before using the output
Revision goal	What you are trying to improve after run 1

B. Appendix B | Prompt quality rubric for the studio

Criterion	Score band	What strong work looks like
Stage discipline	0–2	The prompt stays in its lane and does not jump ahead to later valuation steps
Source discipline	0–2	The packet is named explicitly and outputs carry usable source anchors
Output design	0–2	The schema creates a table or checklist that can go into a workbook or memo process
Fact / inference separation	0–2	Facts, interpretation, and open questions are separated clearly
Auditability	0–2	Another analyst could understand how the prompt affected the work and why the revision improved it
Total	**10**	**A strong prompt package usually scores 8–10 because it is narrow, useful, and revisable**

C. Appendix C | Compact workshop Prompt_Log

Stage	Prompt version	Packet used	What failed or felt weak	Revision made	Why the revision improved the artifact
Architecture	v1				
Architecture	v2				
Fact Bank	v1				
Fact Bank	v2				
Forecast Bridge	v1				

Stage	Prompt version	Packet used	What failed or felt weak	Revision made	Why the revision improved the artifact
Forecast Bridge	v2				
Valuation Control	v1				
Valuation Control	v2				

D. Appendix D | Suggested instructor debrief questions

Questions that reveal whether students own the method

1. Which revision improved your output more: a narrower packet or a tighter schema? Why?
2. In which exercise did the model most strongly try to jump ahead of the stage boundary?
3. What did your best prompt **forbid** that your first prompt allowed?
4. Which stage was hardest to keep auditable: retrieval, bridge-building, or memo writing?
5. What would you change before adapting the workshop method to the final project target company?