

# 10 Frameworks That Unlock Strategic AI for Every Role

## Framework 1: RACE

---

### 1. Context: Why Prompting Matters

Prompting is the act of giving instructions to an AI language model to generate useful, relevant responses. In a world where LLMs can do everything from summarizing documents to simulating investor calls, the *quality of your output depends on the structure of your input*.

Poor prompts often yield vague, verbose, or irrelevant results. Prompt frameworks like **RACE** offer structure and clarity, especially in **sprint planning**, **user story drafting**, or **bug triaging**—where alignment and expectations are critical.

---

### 2. Framework Overview

- **Name:** RACE
  - **Purpose:** A structured prompting framework for instructing AI to generate actionable outputs in fast-paced, goal-driven contexts (e.g., product tickets, standup prep, customer response drafts).
  - **High-Level Structure:**
    - **Role** – Who should the AI act as?
    - **Action** – What needs to be done?
    - **Context** – Background or constraints
    - **Explanation** – Desired output or format
- 

### 3. Step-by-Step Breakdown

<b>Component</b>	<b>Purpose</b>	<b>Example</b>
<b>R - Role</b>	Sets perspective or persona	"Act as a Product Manager at a SaaS company"
<b>A - Action</b>	Defines the task	"Write a user story for a feature request"
<b>C - Context</b>	Supplies necessary background	"The feature involves enabling dark mode in the mobile app"
<b>E - Explanation</b>	Describes expected output	"Format the response as a JIRA ticket with title, description, and acceptance criteria"

---

### 4. Example Prompts & Iterations

### Basic Prompt (Before):

"Write a user story for a dark mode feature."

### Likely Output:

A generic user story with no company context, no formatting, and vague language.

---

### Improved Prompt using RACE:

#### TEMPLATE:

You are a [Role: Product Manager at a SaaS company].

[Action: Draft a user story] for a new feature.

[Context: The feature enables dark mode on the mobile app to reduce eye strain during nighttime use. It should align with existing design tokens.]

[Explanation: Format the output as a JIRA ticket with the following structure: Title, Description, Acceptance Criteria in bullet points.]

#### Expected Output:

**Title:** Enable Dark Mode in Mobile App

**Description:** As a user, I want to enable dark mode in the app so I can reduce eye strain when using the app at night.

#### Acceptance Criteria:

- Toggle to switch between light and dark mode
  - Must follow existing theme token definitions
  - Dark mode should persist across app sessions
- 

## 5. Best Practices (RACE)

- **Be clear about the Role:** Helps the model stay grounded in relevant tone/language.
  - **Use explicit output formatting:** Always specify structure (e.g., bullet points, markdown).
  - **Avoid assumption gaps:** If users or constraints are specific, say so.
  - **Check for verbosity:** RACE doesn't require long answers—just structured ones.
- 

## 6. Use Cases & Generalization

- **Product Management:** Writing user stories, sprint goals, tickets
- **Customer Support:** Generating templated but context-aware replies
- **Content Operations:** Creating briefs, social copy, release highlights
- **Limitation:** RACE is more task-oriented; less suited for explorative or ideation tasks like brainstorming or storytelling.

## Framework 2: COAST

---

### 1. Context: Why Prompting Matters

When delegating multi-step analytical tasks—be it product research, competitive benchmarking, or strategy formulation—AI often fumbles if the prompt lacks structure. That’s because most real-world problems involve more than one objective, and the model needs to know *what’s relevant, what’s expected, and in what order*.

The **COAST** framework shines in such scenarios. It provides a scaffolded instruction format that helps AI **sequence complex tasks**, making it especially powerful for PMs, researchers, analysts, and operations folks working on nuanced investigations.

---

### 2. Framework Overview

- **Name:** COAST
  - **Purpose:** To structure prompts for multi-step, context-rich tasks—especially those involving **analysis, research, or scenario planning**.
  - **High-Level Structure:**
    - **Context** – Give product/market background
    - **Objective** – Define what success looks like
    - **Actions** – List steps that need to be performed
    - **Scenario** – Frame the problem or hypothesis
    - **Task** – Clarify the AI’s final deliverable
- 

### 3. Step-by-Step Breakdown

COMPONENT	PURPOSE	EXAMPLE
<b>C - CONTEXT</b>	Grounds the model in the background	"We're launching a new edtech platform for working professionals."
<b>O - OBJECTIVE</b>	Sets the goal	"Understand market expectations from career-focused edtech products."
<b>A - ACTIONS</b>	Breaks down process	"Analyze competitors, summarize user reviews, identify 3 unmet needs."
<b>S - SCENARIO</b>	Frames use-case with situational detail	"Assume we're launching in India with a \$20/month price point."
<b>T - TASK</b>	Defines the desired output	"Generate a brief with bullet points summarizing findings + recommendations."

---

### 4. Example Prompts & Iterations

**Basic Prompt (Before):**

“Research the edtech market.”

**Likely Output:**

A generic and high-level overview with no target audience, country context, or actionable insights.

---

**Improved Prompt using COAST:**

**TEMPLATE:**

[Context]: We're building a new edtech platform targeted at working professionals seeking to upskill in AI/ML.

[Objective]: Identify unmet user needs and differentiators in the career upskilling space.

[Actions]:

1. List top 5 competitors in the Indian market.
2. Summarize reviews to extract pain points.
3. Identify at least 3 unmet user needs.
4. Suggest potential features based on insights.

[Scenario]: Assume a pricing model of ₹1,500/month and online-only delivery.

[Task]: Present findings in bullet points, ending with 3 feature recommendations.

**Expected Output:**

- **Top Competitors:** UpGrad, Scaler, Great Learning, Coursera, edX
  - **Pain Points:** High prices, inflexible schedules, lack of personalized mentorship
  - **Unmet Needs:**
    - Shorter-duration, focused content (e.g., 4-week sprints)
    - Career mentorship beyond course material
    - Real-world projects with Indian companies
  - **Recommendations:**
    - Weekly 1:1 mentor sessions
    - Live projects with industry tie-ups
    - Affordable tiered pricing for students and job switchers
- 

**5. Best Practices (COAST)**

- Break instructions into **bullet points** so the model parses complex requests step-by-step
  - **Scenario framing** can drastically improve specificity (location, pricing, user segment)
  - **Don't skip the Task line:** Tell the model *how to format* and *what to deliver*
- 

**6. Use Cases & Generalization**

- **Competitive research:** Market comparison, pricing studies, SWOT
- **Growth planning:** Analyzing feedback loops, virality potential, retention drivers
- **UX/Feature ideation:** Synthesizing user interviews into roadmap features
- **Ops Strategy:** Vendor benchmarking, location analysis, pricing simulations
- **Limitations:** May not work as well for highly creative outputs (like storytelling, branding language) unless combined with another technique (e.g., RISE or TAG)

## Framework 3: CREO

---

### 1. Context: Why Prompting Matters

In product management, marketing, research, and UX, you'll often find yourself needing structured outputs—be it user journey maps, stakeholder briefs, or hypothesis summaries. A common pitfall? Vague prompts that lead to unstructured, bloated, or irrelevant outputs.

The **CREO** framework solves this by helping AI lock onto the essentials—**what is the job, why it matters, how it should be done, and what it should produce.**

---

### 2. Framework Overview

- **Name:** CREO
  - **Purpose:** Designed for tasks that require **cleanly formatted, actionable output**—especially useful in documentation, structured writing, analysis, and planning tasks.
  - **High-Level Structure:**
    - **Context** – Set the background or environment
    - **Request** – Define the specific job to be done
    - **Explanation** – Give clarifications, examples, or constraints
    - **Outcome** – Describe the expected format/result
- 

### 3. Step-by-Step Breakdown

COMPONENT	PURPOSE	EXAMPLE
<b>C - CONTEXT</b>	Ground the model with situational info	"We're designing a user onboarding flow for a fitness app"
<b>R - REQUEST</b>	Clearly state what you want	"Create a step-by-step onboarding journey"
<b>E - EXPLANATION</b>	Add instructions, format needs, tone, length	"Make it friendly, mobile-first, include CTAs for each step"
<b>O - OUTCOME</b>	Define output format or deliverable type	"Present in bullet points, structured by screen"

---

## 4. Example Prompts & Iterations

### Basic Prompt (Before):

“Write a user onboarding flow for a fitness app.”

### Likely Output:

Disjointed ideas or a paragraph lacking structure, tone, or platform considerations.

---

### Improved Prompt using CREO:

#### TEMPLATE:

[Context]: We are building a mobile fitness app aimed at young professionals trying to build consistent workout habits.

[Request]: Draft a user onboarding flow.

[Explanation]: The onboarding should take no more than 4 screens. Use a friendly, motivating tone. Include sample CTAs and messages.

[Outcome]: Present output in bullet points, labeled per screen (e.g., Screen 1: Welcome screen, etc.).

### Expected Output:

- **Screen 1: Welcome Screen**
  - Message: “Let’s get you moving, [Name]!”
  - CTA: “Let’s Go 🚀”
- **Screen 2: Goal Setup**
  - Message: “What’s your main fitness goal?”
  - Options: Weight loss, Muscle gain, Flexibility
  - CTA: “Continue”
- **Screen 3: Scheduling**
  - Message: “Choose your preferred workout days”
  - Interactive calendar + notification opt-in
  - CTA: “Set My Routine”
- **Screen 4: Personalization**
  - Message: “One last step: Pick your workout style”
  - Options: Cardio, HIIT, Yoga
  - CTA: “Finish Setup ”

---

## 5. Best Practices (CREO)

- Use “Request” to lock the task—avoid open-ended instructions
  - Always specify “Outcome”—this reduces hallucinations or irrelevant tangents
  - Use tone/style cues in “Explanation”—especially for content tasks
- 

## 6. Use Cases & Generalization

- User flows & onboarding screens
- Email copy, social media templates, release notes
- Pitch decks & PRD modules (FAQs, problem statements, etc.)
- Customer journey stages (mapped with actions/goals)
- Limitations: May not perform well for idea generation or strategic reasoning unless blended with frameworks like RISE or COAST

## Framework 4: CARE

---

### 1. Context: Why Prompting Matters

When communicating with users, executives, or cross-functional teams, the goal is rarely *just* clarity—it’s empathy, alignment, and relevance. Poorly framed prompts lead to robotic, tone-deaf, or overly verbose content that fails to inform or persuade.

The **CARE** framework addresses this by balancing **strategic clarity** with **emotional intelligence**. It’s perfect for PMs drafting UX copy, stakeholder briefs, onboarding messages, or even user persona definitions.

---

### 2. Framework Overview

- **Name:** CARE
- **Purpose:** To help AI produce **user- or stakeholder-focused content** that is clear, concise, actionable, and emotionally attuned.
- **High-Level Structure:**
  - **Context** – Define the audience and background
  - **Action** – State what you need from the model
  - **Result** – Specify what success looks like
  - **Example** – Provide a sample/format for reference

---

### 3. Step-by-Step Breakdown

COMPONENT	PURPOSE	EXAMPLE
C - CONTEXT	Ground the prompt in audience and use-case	"This is for onboarding new users of a B2B analytics platform."
A - ACTION	Define what the model should do	"Write a welcome email with setup instructions."
R - RESULT	Clarify success metrics	"It should drive users to log in and complete setup."
E - EXAMPLE	Share a sample structure or tone	"Follow the tone of Intercom emails—friendly but professional."

---

### 4. Example Prompts & Iterations

#### Basic Prompt (Before):

"Write a welcome email for a SaaS product."

#### Likely Output:

Generic subject line, vague CTA, and no product-specific guidance.

---

#### Improved Prompt using CARE:

##### TEMPLATE:

[Context]: You're a UX writer for a B2B analytics platform that just onboarded new customers post-sales.

[Action]: Write a welcome email that encourages them to log in and begin setup.

[Result]: The email should drive a 70% login rate and minimize drop-off before setup.

[Example]: Use a tone similar to Intercom or Notion welcome emails—conversational, confident, not overly formal. Include CTA buttons like "Start Setup."

#### Expected Output:

**Subject:** Welcome aboard, [First Name]!

Hi [Name],

We're thrilled to have you with us at **InsightHub**. Your dashboard is ready and waiting.

To get the most out of InsightHub, start by setting up your team, defining KPIs, and connecting your data source. It only takes 10 minutes.

#### Start Setup

If you need help, we're here 24/7.

— Team InsightHub

---

## 5. Best Practices (CARE)

- Always include a **sample or tone reference** in the Example section.
- Don't skip **Result**—this tells the model what “good” looks like (clicks, engagement, clarity).
- CARE works best when the audience is defined—add persona labels like “new user,” “executive,” “ops analyst.”

---

## Use Cases & Generalization

- **UX Writing:** Empty state messages, nudges, onboarding flows
- **Internal Briefs:** For design, marketing, engineering
- **Customer Messaging:** Emails, notifications, support updates
- **Executive Summaries:** Strategy updates or recommendations
- **Limitations:** Not ideal for deeply technical tasks like debugging, or highly visual tasks like UI layout suggestions

## Framework 6: PAIN

---

### 1. Context: Why Prompting Matters

In high-stakes operational or technical scenarios—think outages, support escalations, or post-launch issues—clarity and structure are critical. A generic prompt like “What went wrong here?” yields vague, unhelpful outputs.

The **PAIN** framework provides a dependable prompt scaffold for instructing AI to think diagnostically, identify key data, and recommend actionable follow-ups.

---

### 2. Framework Overview

- **Name:** PAIN
- **Purpose:** A diagnostic prompt framework for **analyzing issues and generating resolution plans** in technical or business contexts.
- **High-Level Structure:**
  - **Problem** – Describe the issue clearly
  - **Action** – Define necessary steps or investigation areas
  - **Information** – Provide supporting data, logs, or examples
  - **Next Steps** – Request a structured recovery or follow-up plan

---

### 3. Step-by-Step Breakdown

COMPONENT	PURPOSE	EXAMPLE
<b>P - PROBLEM</b>	Articulate the issue at hand	"Users are unable to save progress in the mobile app"
<b>A - ACTION</b>	Clarify the expected analysis/fix	"Identify likely root causes and impacted systems"
<b>I - INFORMATION</b>	Give supporting details	"Errors spiked post-deployment (log ID: 1045), rollback happened 2 hours later"
<b>N - NEXT STEPS</b>	Ask for follow-up or mitigation plan	"List short-term fixes and long-term action items"

---

### 4. Example Prompts & Iterations

#### Basic Prompt (Before):

"Figure out what caused the mobile app to crash."

#### Likely Output:

Speculative reasoning with no structure, mitigation plan, or action priority.

---

#### Improved Prompt using PAIN:

##### TEMPLATE:

[Problem]: After the latest Android app update, users are reporting that their progress isn't saving. This started occurring around 4:30 PM IST.

[Action]: Identify potential causes for the data loss and what systems (e.g., cache, API, DB) may be affected.

[Information]:

- Rollout version: v3.2.1
- Error logs show spike in "400 - Bad Request" responses to Save API
- DB latency also increased during that window

[Next Steps]: Suggest a short-term workaround and a detailed list of actions for post-mortem review.

#### Expected Output:

##### Root Cause Hypothesis:

Likely caused by schema mismatch in Save API v2 and client payload structure.

##### Impacted Systems:

- Android client (v3.2.1)
- Save API

- MongoDB write operations

#### Short-Term Fix:

- Force rollback of client to v3.1.9
- Suppress Save API v2 calls

#### Long-Term Action Items:

- Add schema validation layer
  - Introduce regression tests for Save API
  - Update deployment playbook with rollback triggers
- 

### 5. Best Practices (PAIN)

- The **Information** section can include logs, metrics, timestamps, or user symptoms—always be specific.
  - Ask for **step-wise recovery plans** in **Next Steps**, not just a fix.
  - Use **incident ticket IDs or version numbers** for traceability.
- 

### 6. Use Cases & Generalization

- **Incident response & RCA**
- **Support ticket escalations**
- **System reliability post-mortems**
- **CX breakdown analysis**
- **Team retrospectives or service health reviews**
- **Limitations:** PAIN is analytical—it's not designed for ideation, creative writing, or long-term strategy prompts.

## Framework 7: RISE

---

### 1. Context: Why Prompting Matters

High-level planning often requires AI to think like a cross-functional team member—balancing context, inputs, and execution clarity. Without a structured prompt, LLMs tend to generate verbose suggestions with little strategic depth.

The **RISE** framework brings **clarity, hierarchy, and actionability** to prompts used for **roadmaps, feature prioritization, or go-to-market** strategies.

---

### 2. Framework Overview

- **Name:** RISE

- **Purpose:** A prompt structure for **role-based, input-driven, step-by-step execution** of strategic product and planning tasks.
- **High-Level Structure:**
  - **Role** – Define who the model should act as
  - **Input** – Provide background, data, or artifacts
  - **Steps** – Request a logical breakdown of the approach
  - **Execution** – Ask for the final deliverable or output

### 3. Step-by-Step Breakdown

COMPONENT	PURPOSE	EXAMPLE
<b>R - ROLE</b>	Guides tone and decision-making lens	"You are a Senior Product Manager at a B2B SaaS company"
<b>I - INPUT</b>	Context, data, goals	"We're planning Q3 roadmap for our analytics product, focusing on churn reduction"
<b>S - STEPS</b>	Frame how the response should unfold	"Break down the roadmap planning process into clear phases"
<b>E - EXECUTION</b>	Deliver structured output	"Provide the prioritized list of initiatives and a rationale for each"

### 4. Example Prompts & Iterations

#### Basic Prompt (Before):

"Create a product roadmap for Q3."

#### Likely Output:

A generic, unordered list with no context or alignment to business goals.

#### Improved Prompt using RISE:

##### TEMPLATE:

[Role]: You are a Senior Product Manager at a B2B SaaS company specializing in analytics.

[Input]: Our Q3 objective is to reduce customer churn, especially among mid-tier clients. Customer interviews highlight a lack of actionable insights and difficult onboarding as key pain points.

[Steps]:

1. Identify 3 initiatives that could improve retention
2. Estimate effort and potential impact for each
3. Prioritize them based on effort-impact tradeoff

[Execution]: Output should be a roadmap table with Initiative, Goal, Effort (Low/Med/High), Impact (Low/Med/High), and Rationale.

**Expected Output:**

INITIATIVE	GOAL	EFFORT	IMPACT	RATIONALE
GUIDED ONBOARDING FLOW	Improve activation & usage	Medium	High	40% of churned users dropped off within 3 days
PERSONALIZED EMAIL INSIGHTS	Deliver recurring value	Low	Medium	Addresses “lack of insights” issue without heavy dev work
USAGE-BASED RETENTION NUDGES	Alert at-risk users	Medium	Medium	Prevents silent drop-off via engagement signals

**5. Best Practices (RISE)**

- The **Role** should reflect the decision-maker's POV (PM, designer, strategist, etc.)
- In **Steps**, be explicit about sequencing: ideation → prioritization → formatting
- For **Execution**, always ask for structure (table, bullets, JSON) and *why*

**6. Use Cases & Generalization**

- **Product roadmap creation**
- **Strategic feature planning**
- **Go-to-market rollouts**
- **Experiment prioritization**
- **Team OKRs / quarterly planning**
- **Limitations:** RISE is structure-heavy—less useful for free-flowing brainstorming or user-generated content

**Framework 8: ROSES**

**1. Context: Why Prompting Matters**

In collaborative or cross-functional efforts like GTM planning or sprint retros, AI often struggles when the prompt lacks situational clarity or phased reasoning. You may get lists or commentary, but without actionable structure.

The **ROSES** framework addresses this by aligning AI output with real-world strategic thinking. It blends **persona assignment, scenario framing, and expected solution design**—a great fit for fast-paced product environments.

---

## 2. Framework Overview

- **Name:** ROSES
  - **Purpose:** To help the model produce **role-aware, outcome-oriented, and step-structured plans**, particularly for product planning, strategy recaps, and retrospectives.
  - **High-Level Structure:**
    - **Role** – Assign a stakeholder persona
    - **Objective** – Clarify the desired goal
    - **Scenario** – Define context or challenge
    - **Expected Solution** – Frame the type of output needed
    - **Steps** – Guide the AI to think through the solution process
- 

## 3. Step-by-Step Breakdown

COMPONENT	PURPOSE	EXAMPLE
<b>R - ROLE</b>	Provides a realistic voice/perspective	"Act as a Product Marketing Manager preparing a GTM plan"
<b>O - OBJECTIVE</b>	Establishes clear success criteria	"Ensure 40% feature adoption in the first 30 days"
<b>S - SCENARIO</b>	Adds situational nuance	"The feature is launching to SMB users amid peak campaign season"
<b>E - EXPECTED SOLUTION</b>	Defines format and boundaries	"Generate a 5-point GTM checklist"
<b>S - STEPS</b>	Ensures a logical flow of action	"Think through messaging, collateral, channels, enablement, and success tracking"

---

## 4. Example Prompts & Iterations

### Basic Prompt (Before):

"Create a go-to-market plan for a new feature."

### Likely Output:

Unstructured suggestions with unclear priority or alignment to user base.

---

### Improved Prompt using ROSES:

### TEMPLATE:

[Role]: Act as a Product Marketing Manager launching a new AI summarization feature inside our B2B analytics product.

[Objective]: Drive 40% user adoption within 30 days post-launch.

[Scenario]: The launch is scheduled during our annual campaign season. Existing users are mid-market decision-makers.

[Expected Solution]: Generate a GTM checklist covering key elements across messaging, sales enablement, and engagement.

[Steps]:

1. Define value proposition and positioning
2. Suggest launch messaging per channel
3. Recommend sales/CS enablement collateral
4. Propose in-app or email nudges
5. Define basic adoption metrics

**Expected Output:**

#### **GTM Launch Checklist**

1. **Value Prop:** "Save 6+ hours weekly with instant data summaries."
2. **Messaging:**
  - Email: Focus on time saved for analysts
  - In-app banner: "Try 1-click summaries on your dashboard"
3. **Enablement:**
  - Sales: One-pager with use cases + objections
  - CS: Demo video for first-line support
4. **Engagement:**
  - Post-login tooltips for new users
  - Email reminder 7 days post non-use
5. **Metrics:**
  - % of DAUs triggering summary
  - Repeat usage within first week
  - NPS delta from non-users

---

#### **5. Best Practices (ROSES)**

- Think of ROSES as a **sprint brief + retrospective tool**—it's as much about execution as it is about reflection.
- Always define **Expected Solution** to shape the format
- In **Scenario**, include constraints (e.g., time, audience, environment)

---

## 6. Use Cases & Generalization

- **Sprint Planning & Weekly Retros**
- **Feature Launches / GTM Readiness**
- **Quarterly Enablement Briefs**
- **Customer Journey Mapping (e.g., identify friction zones)**
- **Limitations:** Less ideal for creative ideation or long-form copywriting; combine with CREO or TAG when needed

## Framework 9: APE

---

### 1. Context: Why Prompting Matters

When you're dealing with tactical, fast-turnaround tasks—like summarizing feedback, drafting FAQs, or generating simple analysis—a prompt that's too open-ended often results in bloated or misaligned responses.

The **APE** framework helps keep the model **focused, contextual, and output-ready**. It's all about **clarity of intent, brevity in execution, and alignment with business needs**—perfect for busy product, support, or research teams.

---

### 2. Framework Overview

- **Name:** APE
  
- **Purpose:** To quickly and efficiently generate tactical outputs with minimal setup—perfect for operational and repetitive content generation.
- **High-Level Structure:**
  - **Action** – State the job clearly
  - **Purpose** – Define the outcome or business value
  - **Execution** – Guide the formatting or process

---

### 3. Step-by-Step Breakdown

COMPONENT	PURPOSE	EXAMPLE
A - ACTION	Clarify the task	"Generate 5 FAQs for our pricing page"
P - PURPOSE	Tie the task to a business outcome	"Help reduce bounce rate and inbound queries"
E - EXECUTION	Define how it should be done	"Use short, confident answers with a friendly tone. Bullet format."

---

### 4. Example Prompts & Iterations

#### Basic Prompt (Before):

"Write some FAQs for our product."

#### Likely Output:

Random questions, inconsistent tone, or irrelevant focus areas.

---

#### Improved Prompt using APE:

##### TEMPLATE:

[Action]: Generate 5 FAQs for the pricing section of our B2B SaaS product's landing page.

[Purpose]: These FAQs should pre-empt pricing objections and reduce drop-offs or support tickets from new prospects.

[Execution]: Keep answers under 30 words. Use a confident and helpful tone. Present in Q&A bullet format.

#### Expected Output:

##### Q1: Are there any setup fees?

A: No setup fees—just pick a plan and go.

##### Q2: Can I switch plans later?

A: Absolutely. You can upgrade or downgrade anytime.

##### Q3: Is support included in all plans?

A: Yes, live chat and email support are included.

##### Q4: Do you offer volume discounts?

A: Yes, reach out to sales for custom pricing.

##### Q5: What's the cancellation policy?

A: Cancel anytime. No contracts. No hassle.

---

## 5. Best Practices (APE)

- Use **APE for utility-first content**—FAQ, summaries, tweets, support macros, onboarding text
  - Always define the **Purpose**—even if the Action is clear, the business reason helps shape tone
  - In **Execution**, specify constraints: word count, format, tone
- 

## 6. Use Cases & Generalization

- **FAQs and Help Center content**
- **User review summarization**
- **Support ticket macros or canned responses**
- **Product feature descriptions (in short)**
- **Quick response briefs or stakeholder summaries**
- **Limitations:** Too lightweight for complex tasks or strategic thinking—pair with RISE or ROSES for planning work

## Framework 10: CREATE

---

### 1. Context: Why Prompting Matters

LLMs are powerful assistants for drafting structured documents—but only when they know **who to write as, what to write, and how to present it**. Without that guidance, outputs can be verbose, inconsistent, or miss the mark.

The **CREATE** framework solves this by guiding the AI through six core dimensions: voice, task, examples, constraints, format, and extra requirements. It's ideal for **professional writing, structured reporting, and long-form prompt composition**.

---

### 2. Framework Overview

- **Name:** CREATE
- **Purpose:** To design comprehensive prompts for tasks that require a combination of **clarity, tone, structure, and adaptability**, particularly in document generation.

- **High-Level Structure:**

- **Character** – Assign voice or persona
- **Request** – Define the job
- **Examples** – Provide references or previous samples
- **Adjustments** – Offer constraints or style tweaks
- **Type** – Specify output format
- **Extras** – Add any additional instructions or metadata

### 3. Step-by-Step Breakdown

COMPONENT	PURPOSE	EXAMPLE
<b>C - CHARACTER</b>	Defines persona, tone, or role	"You are a senior product manager writing for exec stakeholders"
<b>R - REQUEST</b>	The core task	"Draft a PRD for a new collaboration feature"
<b>E - EXAMPLES</b>	Reference past outputs or formats	"Refer to our previous PRD on 'Team Mentions' for tone and structure"
<b>A - ADJUSTMENTS</b>	Apply style, tone, word count, or angle	"Keep it concise (under 800 words), use business-focused tone"
<b>T - TYPE</b>	Define the format (bullet list, doc, slide outline)	"Write in markdown format with clear headers"
<b>E - EXTRAS</b>	Add anything custom (e.g., add links, tags, disclaimers)	"Include 3 user stories and relevant JIRA tag IDs"

### 4. Example Prompts & Iterations

**Basic Prompt (Before):**

“Write a PRD for a new Slack-like messaging feature.”

**Likely Output:**

An unstructured wall of text with unclear goals, audience, or formatting.

**Improved Prompt using CREATE:**

**TEMPLATE:**

[Character]: You are a Senior Product Manager at a productivity SaaS company. Write in a business-executive tone.

[Request]: Draft a PRD for a new feature that enables inline comments in collaborative docs.

[Examples]: Use our previous PRD on “real-time cursor tracking” as a reference for structure and clarity.

[Adjustments]: Keep the length under 800 words. Emphasize user impact and design constraints.

[Type]: Use markdown format with sections like: Problem, Goals, Success Metrics, User Stories, Dependencies.

[Extras]: Include at least 3 detailed user stories and embed mock link placeholders in the “Design” section.

### **Expected Output (Excerpt):**

#### **Problem**

Users cannot annotate or discuss document content without interrupting the flow of writing. This creates bottlenecks in async collaboration.

#### **Goals**

- Enable inline comments across all document types
- Maintain WYSIWYG editing flow
- Real-time notifications for mentions

#### **Success Metrics**

- 80% of users use inline comments in first 2 weeks
- 25% reduction in Slack messages referencing document discussions

#### **User Stories**

1. As a user, I want to highlight text and add a comment that others can reply to.
2. As a team member, I want to be notified when I’m mentioned in a comment.
3. As an admin, I want to control who can comment or resolve threads.

---

## **5. Best Practices (CREATE)**

- This is your **go-to framework for long-form AI tasks**. Use it for docs, decks, support guides, release notes, etc.
- **Provide reference materials** or examples when possible—it drastically improves quality.
- Don’t forget to define **"Extras"**—tags, metadata, visuals, etc., are crucial in team settings.

---

## **6. Use Cases & Generalization**

- **Product Requirement Documents (PRDs)**
- **Release Notes & Help Articles**
- **Technical Briefs & Knowledge Base Drafts**
- **Internal Memos, Onboarding Docs, Meeting Recaps**
- **Workshop Summaries or Sprint Planning Docs**

- **Limitations:** Can be verbose if “Adjustments” and “Type” are not well defined