Corcava

Run Projects From Chat Playbook

Telegram-first workflows for task capture, daily digests, and overdue escalation.

What this is

A practical playbook for managing tasks, reminders, and updates in Telegram without losing visibility or accountability.

Quick links

Template URL	https://corcava.com/templates/chat-ops/run-projects-from-chat-playbook	**Scan**
Related templates	Telegram Task Bot Commands Slack to Telegram Integration Checklist	Open on mobile

Version: January 2026

How to use this playbook

Pick one workspace chat as the default (Telegram), then implement the three workflows in order. Start simple, measure noise, and only then add automation or routing rules.

- **Workflow 1:** Task intake (capture, clarify, assign, confirm).
- **Workflow 2:** Daily digest (one summary message that drives the day).
- **Workflow 3:** Overdue escalation (predictable nudges, not surprise pings).

What you'll need

- One Telegram chat that everyone agrees is the default for work capture (a team chat or project chat).
- A single owner for triage (rotating is fine) and a single source of truth for task status.
- A quiet-hours policy (even a simple one).

Workflows at a glance

Workflow	Goal	Output
Task intake	Turn chat messages into assigned work with due dates	A task with owner + due date + definition of done
Daily digest	Keep priorities visible without constant back-and-forth	A single daily message with priorities, due soon, blockers
Overdue escalation	Prevent silent slippage	Timed nudges that escalate only when needed

Rules of the road

The fastest way to fail at Chat Ops is to let chat become both the inbox and the database. Use chat for capture and coordination, and keep status in one place.

Recommended defaults

- **One inbox:** Decide where tasks are captured (Telegram) and stick to it.
- **One truth:** Task status lives in one system, not across multiple chats.
- **Short messages:** One task per message. Use a consistent format.
- **Time boundaries:** Quiet hours + escalation rules reduce anxiety and noise.
- **Decision logging:** Any decision that changes scope or priority gets a short log entry.

Chat hygiene rules (copy/paste)

Rule	Why it matters	Example
One task per message	Makes capture and assignment reliable	Create landing page wireframe - due Thu - owner @Sam
Define done	Prevents loops and rework	Done = approved copy + final export + uploaded to portal
No ambiguous deadlines	Avoids ASAP and hidden urgency	Use a date or end of day in a timezone
Escalate on a timer	No surprise pings, predictable behavior	Warn 24h before due, escalate 24h after overdue

Workflow 1: Task intake (capture to assignment)

Use this when tasks arrive in chat, email, or meetings and you need a fast way to prevent loss. The output of this workflow is a task with owner, due date, and definition of done.

Steps

- **Capture:** Post the task in Telegram using a standard template.
- **Clarify:** Add missing inputs (client, due date, definition of done, dependencies).
- **Assign:** One person owns the next action (even if others help).
- **Confirm:** The owner replies with a short acknowledgement and next step.
- **Track:** Status updates happen in the task system; chat gets only key updates.

Message templates (placeholders)

Template A: Capture

```
New task: [what]
Client/Project: [name]
Owner: @[name]
Due: [date + timezone]
Definition of done: [1 sentence]
```

Template B: Clarify

```
Need clarification: [question]
Options: [A] / [B]
Decision needed by: [date]
Blocking: [yes/no]
```

Intake checklist

Item	Yes/No	Notes
Owner assigned		
Due date + timezone set		
Definition of done written		
Dependencies noted		
Client visibility decided (internal vs client-facing)		

Workflow 2: Daily digest (one message that drives the day)

A digest replaces scattered updates. It should be short enough to read in 30 seconds and specific enough to drive action.

Recommended cadence

- **Teams:** Post at the start of the workday (or the handoff point if you span timezones).
- **Client-facing projects:** Post 2–3 times per week rather than daily, unless the work is fast-moving.
- **Rule:** If the digest is longer than 12 lines, split into Today and Risks.

Digest template (copy/paste)

```
Daily digest


[Project] Daily digest - [Date]

Top priorities (3):
1) ...
2) ...
3) ...

Due soon (next 48h):
- ...

Blocked / needs decision:
- ...

New tasks captured:
- ...
```

What to include vs avoid

Include	Avoid
Named owner + next step	Long context threads
Dates and timezones	Vague urgency (ASAP)
Decisions needed (with deadline)	Status that doesn't change actions

Workflow 3: Overdue escalation (predictable, not stressful)

Escalation should feel like a safety net. The goal is to catch slippage early, not to shame people.

Escalation ladder (default)

Timing	Who gets notified	Message goal
24h before due	Owner	Confirm plan + unblock early
Due date (morning)	Owner + triage owner	Confirm next step
24h overdue	Owner + project lead	Decide: re-scope, re-assign, or re-date
72h overdue (or client risk)	Project lead + stakeholder	Escalate decision / reset expectations

Escalation messages (placeholders)

Nudge (24h before due)

```
Reminder: [task] is due [date/time].
Owner: @[name]
Any blockers?
If yes, reply with:
BLOCKED - [reason] - need [input].
```

Reset (24h overdue)

```
[task] is now overdue.

Options:
A) Keep scope, move due date to [new date]
B) Reduce scope to [smaller deliverable] by [date]
C) Re-assign to @[new owner]

Reply with A/B/C by [time].
```

Rollout plan (fast, low-friction)

Implement in three short phases so you can adjust for your team's culture and tolerance for notifications.

Phase 1: 30-minute setup

- Pick the default Telegram chat (team or project).
- Agree on the intake message format and who does triage this week.
- Set quiet hours and escalation ladder (use the defaults above).

Phase 2: Week 1

- Run task intake + daily digest only. Do not add more alerts yet.
- Measure: tasks captured per day, missed tasks, digest length.

Phase 3: Week 2+

- Add overdue escalation with a single warning step (24h before due).
- Only add Slack routing if Telegram is stable and low-noise.

How to map this into Corcava (worksheet)

Use this table to map each playbook element to your Corcava setup. Keep it specific and owned.

Playbook element	In Corcava	Owner	Notes
Task capture from Telegram	[Link/feature: Telegram integration]		Decide which chat(s) are allowed to create tasks
Assignment + due dates	[Task assignment + due date fields]		Define required fields before a task is accepted
Daily digest	[Scheduled report / digest]		Post to one chat, same time daily
Overdue escalation	[Automation / reminders]		Start with 1 warning step; tune quiet hours
Slack to Telegram routing (optional)	[Slack integration / webhook]		Only route high-signal alerts (mentions, overdue, new lead)

Related templates

- Telegram Task Bot Commands Cheat Sheet
- Slack to Telegram Integration Checklist
- Telegram Daily Standup Template
- Project Kickoff Checklist and Agenda