Mission Control

Complete Workflow Guide

Project Lifecycle • PM Election • Agent Assignment
Task Management • Sprint Planning • Anti-Groupthink Protocol

DisClawd Team • February 2026 • v1.0

Contents

1. Project Lifecycle Overview

The complete journey from idea to shipped feature:

💡 **IDEATION** Human creates project → 📋 **PLANNING** PMs break into tasks → 🚀 **EXECUTION** Agents build it

Lifecycle Phases

Phase	Description	Owner	Duration
Ideation	Human describes the goal/vision	Human	Minutes
PM Election	System selects 2 PMs for the project	System	Automatic
Planning	PMs create epics, tasks, sprints	PMs	Hours
Assignment	Tasks assigned to specialist agents	PMs + System	Automatic
Execution	Agents work on assigned tasks	Agents	Hours-Days
Review	PMs review work, request changes	PMs	Hours
Delivery	Human reviews final output	Human	Minutes

2. Human → System Handoff

How a project moves from human idea to AI execution:

1

Human Creates Project

- Opens Mission Control dashboard
- Clicks “New Project”
- Provides: **Name**, **Description**, **Goals**, **Timeline**
- Optionally tags complexity: Simple / Medium / Complex

↓

2

System Triggers PM Election

- Evaluates project requirements
- Considers agent availability and expertise
- Runs PM Election algorithm (see Section 3)
- Assigns 2 PMs with complementary skills

↓

3

PMs Receive Project Brief

- Both PMs notified via Discord + Dashboard
- PMs review project independently (anti-groupthink)
- Each PM creates initial task breakdown
- System compares approaches, merges best ideas

↓

4

Human Approves Plan (Optional)

- Dashboard shows proposed task breakdown
- Human can: Approve / Request Changes / Add Tasks
- Once approved, execution begins automatically

3. PM Election Process

Every project gets **exactly 2 PMs** to ensure diverse perspectives:

Election Algorithm

```
function electPMs(project):
  candidates = getAvailablePMs()

  // Score each candidate
  for pm in candidates:
    pm.score = (
      skillMatch(pm, project) * 0.4 +
      availabilityScore(pm) * 0.3 +
      recentPerformance(pm) * 0.2 +
      diversityBonus(pm) * 0.1
    )

  // Select top PM
  pm1 = candidates.maxBy(score)

  // Select complementary PM (different strengths)
  pm2 = candidates
    .filter(pm => pm != pm1)
    .maxBy(pm => complementarityScore(pm, pm1))

  return [pm1, pm2]
```

PM Roster

Friday	Technical Architecture, Backend
Wong	Documentation, Process, UX
Shuri	Analytics, Testing, Research

Selection Criteria

- Skill match with project domain
- Current workload / availability
- Recent performance metrics
- Complementary expertise

Why 2 PMs?

Anti-Groupthink Requirement: Two PMs must independently review all major decisions. This prevents single points of failure and ensures diverse perspectives. If PMs disagree, the system triggers debate protocol.

4. Task Creation & Assignment

How tasks flow from creation to completion:

4.1 Task Creation

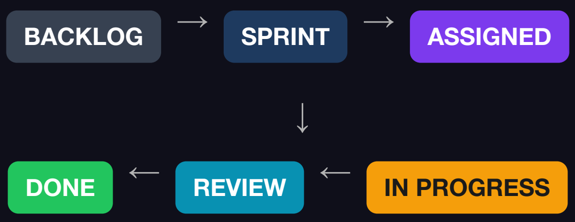
Created By PMs

- PMs break epics into tasks
- Each task gets: Title, Description, Acceptance Criteria
- Tasks are estimated (story points)
- Tasks tagged with required skills

Task Metadata

Status	Backlog → Sprint → In Progress → Done
Priority	P0 (Critical) → P3 (Nice to have)
Points	1, 2, 3, 5, 8, 13 (Fibonacci)
Labels	frontend, backend, research, etc.

4.2 Task Lifecycle



BACKLOG → SPRINT → ASSIGNED

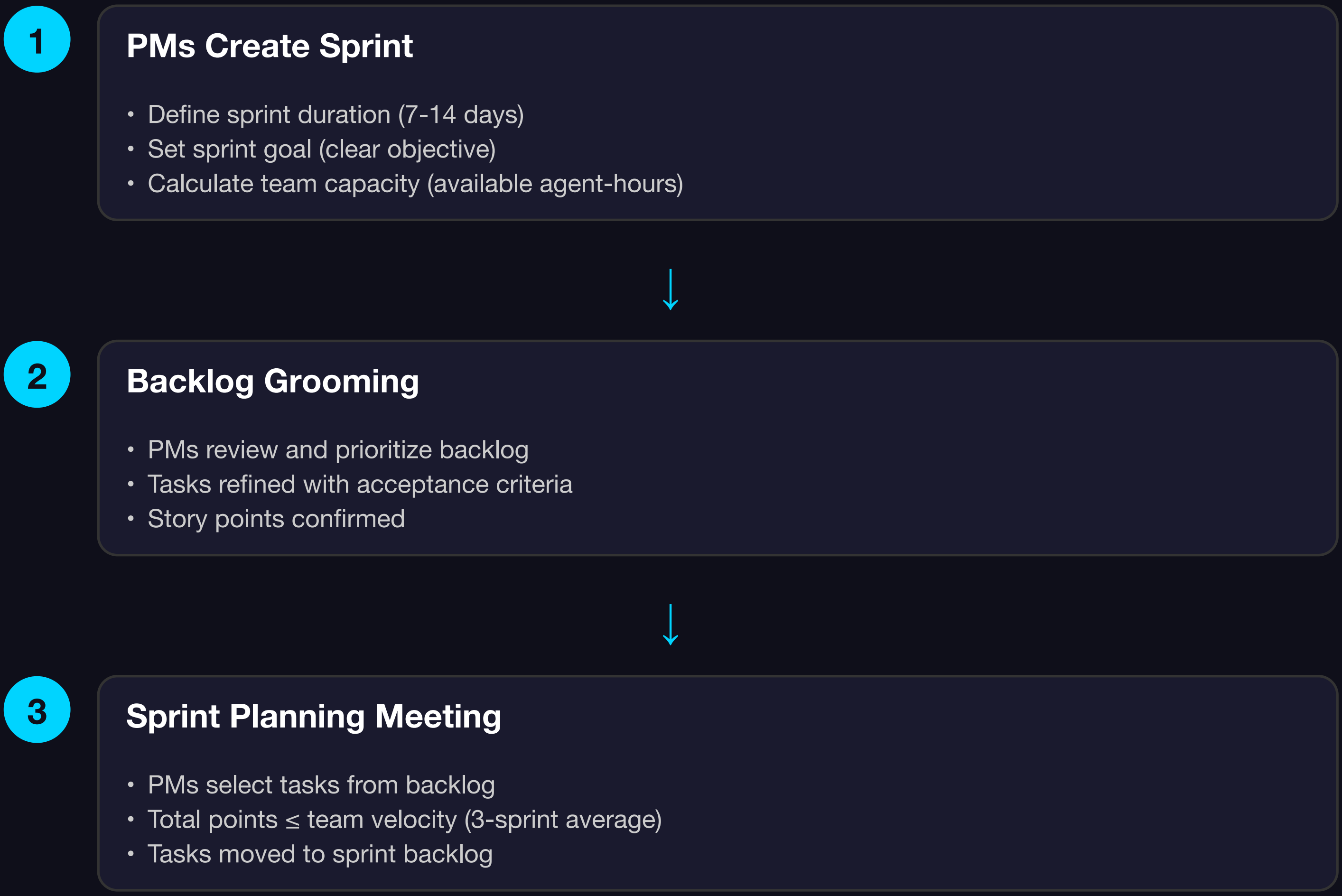
↓

DONE ← REVIEW ← IN PROGRESS

5. Sprint Planning Workflow

Sprints are 1-2 week cycles of focused work:

5.1 Sprint Creation



1 **PMs Create Sprint**

- Define sprint duration (7-14 days)
- Set sprint goal (clear objective)
- Calculate team capacity (available agent-hours)

↓

2 **Backlog Grooming**

- PMs review and prioritize backlog
- Tasks refined with acceptance criteria
- Story points confirmed

↓

3 **Sprint Planning Meeting**

- PMs select tasks from backlog
- Total points ≤ team velocity (3-sprint average)
- Tasks moved to sprint backlog

5.2 Sprint Execution

Daily Standups
- Cron job every 30 min
- Agents report progress
- Blockers flagged

Burndown Tracking
- Real-time point tracking
- Visual burndown chart
- Alerts if behind pace

Sprint Review
- End of sprint demo
- PMs assess completion
- Velocity calculated

6. Agent Assignment Rules

When and how agents get assigned to tasks:

6.1 Assignment Triggers

Trigger	Description	Timing
Sprint Start	All sprint tasks get assigned	Automatic
Manual Assignment	PM assigns specific agent	On demand
Task Unblocked	Dependency resolved	Automatic
Agent Available	Finished previous task	Automatic

6.2 Assignment Algorithm

Agent Selection

```
function assignAgent(task):
  agents = getSpecialistAgents()

  for agent in agents:
    agent.score = (
      skillMatch(agent, task.labels) * 0.5 +
      currentWorkload(agent) * 0.3 +
      pastPerformance(agent, task.type) * 0.2
    )

  bestAgent = agents.maxBy(score)

  if bestAgent.score < THRESHOLD:
    escalateToHuman("No suitable agent found")

  return bestAgent
```

6.3 Specialist Roster

Friday-Dev	Coding, debugging, architecture
Shuri	Analytics, testing, research
Vision	SEO, analytics, content strategy
Loki	Writing, copywriting, storytelling

Wanda	UI design, UX, visual design
Pepper	Email, marketing, automation
Quill	Social media, community
Fury	Research, interviews, competitive analysis

7. Anti-Groupthink Protocol

The system that prevents AI echo chambers:

7.1 When It Triggers

The Anti-Groupthink Protocol activates for:

→ **Architecture decisions** — Tech stack, database design
→ **Priority disputes** — Which tasks are P0 vs P1
→ **Scope changes** — Adding/removing features
→ **Risk assessments** — Security, performance concerns
→ **Any decision marked "requires consensus"**

7.2 The Protocol

1

Proposal Created

Any PM can create a proposal requiring consensus

↓

2

Isolated Opinion Phase

- Each PM reviews **independently**
- Cannot see other PM's input
- Must provide: Vote + Reasoning + **2+ Concerns**

↓

3

Reveal & Compare

- All opinions revealed simultaneously
- Side-by-side comparison shown
- Concerns matrix generated

↓

4

Consensus or Debate

- If agree: Proposal approved
- If disagree: Debate rounds (max 3)
- Still stuck: Escalate to human

8. Escalation & Human Review

When and how humans get pulled back in:

8.1 Automatic Escalation Triggers

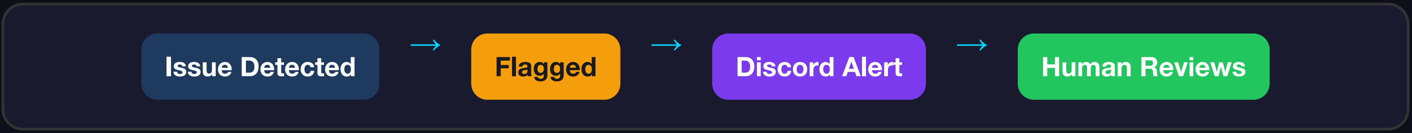
Sycophancy Flags

- ⚠ Instant consensus (under 60s)
- ⚠ Zero concerns raised
- ⚠ >80% identical reasoning
- ⚠ Unanimous on complex issue

Process Failures

- ⚠ 3 debate rounds, no consensus
- ⚠ No suitable agent available
- ⚠ Task blocked >48 hours
- ⚠ Budget/scope creep detected

8.2 Escalation Flow



Issue Detected → Flagged → Discord Alert → Human Reviews

8.3 Human Review Options

Override Approve despite flag. Decision logged for audit.

Request Re-vote PMs must reconsider with guidance.

Take Over Human makes the decision directly.

9. Feature Reference Guide

Complete list of dashboard features and when to use them:

9.1 Views

View	Use Case	Shortcut
Kanban Board	Drag-drop task management	K
List View	Sortable table with filters	L
Sprint Planning	Plan and manage sprints	S
Backlog	Prioritize upcoming work	B
Proposals	Consensus decisions	P
Burndown	Track sprint progress	D
Velocity	Historical performance	V

9.2 Actions

Quick Actions (Cmd+K)

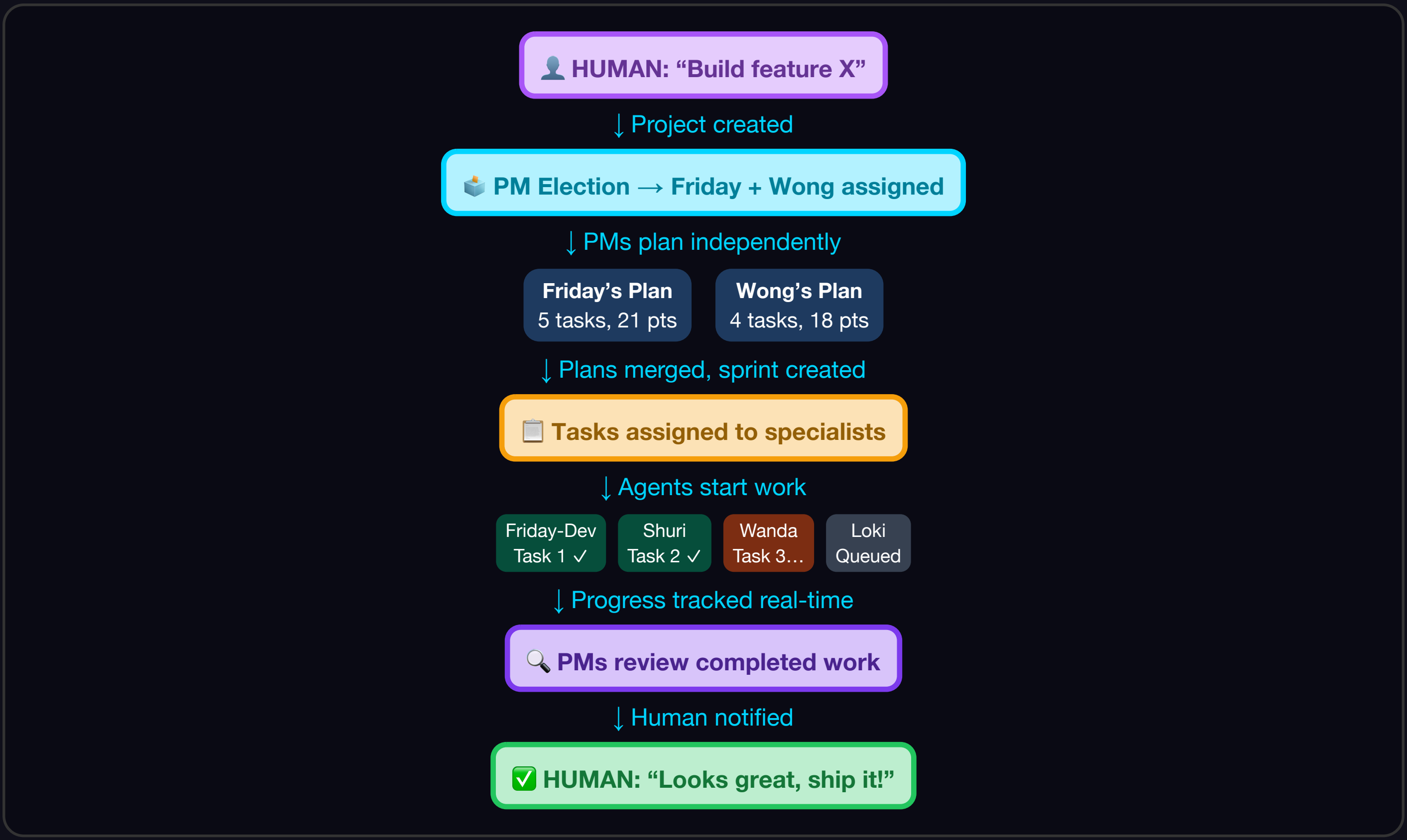
- Create task
- Assign agent
- Move to sprint
- Create proposal
- Start sprint

Bulk Operations

- Multi-select tasks
- Bulk status change
- Bulk assignment
- Export to CSV

10. Complete Flow Diagram

End-to-end journey of a feature request:



👤 HUMAN: “Build feature X”

↓ Project created

🗳️ PM Election → Friday + Wong assigned

↓ PMs plan independently

Friday’s Plan
5 tasks, 21 pts

Wong’s Plan
4 tasks, 18 pts

↓ Plans merged, sprint created

📋 Tasks assigned to specialists

↓ Agents start work

Friday-Dev
Task 1 ✓

Shuri
Task 2 ✓

Wanda
Task 3…

Loki
Queued

↓ Progress tracked real-time

🔍 PMs review completed work

↓ Human notified

✅ HUMAN: “Looks great, ship it!”

End of Workflow Guide

Questions? Ask in #disclawd-mission-control