IRIS-v4 Web3 Upgrade – **Agile Sprint Plan & Deliverables Board**

> A living **Jira-style backlog** that converts the three-volume guide into **time-boxed sprints**.
> Copy each ticket into GitHub issues, assign owners, and ship incrementally.

📌 Project Charter (1-pager)

Field	Value
Vision	Turn iris-v4 into the first **production-ready AI × Web3 agent platform** where any user can spawn an agent that reads Ethereum, signs transactions, mints NFT memories, and accepts USDC payments.
KPIs	• 3 on-chain tools live in web3-mcp • NFT mint flow end-to-end • USDC payment gate working • Flutter wallet connect • 90 % unit-test coverage
Non-Goals	Cross-chain swaps, decentralized GPU (Phase-2 only)

🗓️ Release Road-Map

Release	Sprint #	Theme	Public Demo
Alpha	1-2	Read-only Web3 (balance, tx)	`/demo/balance`
Beta	3-4	Transaction signing + NFT mint	`/demo/nft-mint`
GA	5-6	USDC payments + launchpad UI	Public beta on testnet

🏃 Sprint Backlog (6 × 2-week sprints)

Sprint 1 – **Web3-MCP Scaffold** (Days 1-14)

ID	Title	Definition-of-Done	Points
WEB3-1	Scaffold `mcp-servers/web3-mcp` Dockerfile & CI	Image builds & pushes to GHCR	2
WEB3-2	`GET /get_eth_balance` endpoint	Returns correct balance for 0x address	3
WEB3-3	`GET /get_transaction_receipt` endpoint	JSON matches Etherscan schema	3

ID	Title	Definition-of-Done	Points
WEB3-4	Register web3 tools in `agent_manager.py`	Tools appear in `/tools` list	2
DEVOPS-1	Add `web3-mcp` service to `docker-compose.yml`	Container starts w/ env vars	2

Sprint Goal: Agent can answer "What's my ETH balance?" in chat.

Sprint 2 – **Unsigned Transaction Flow** (Days 15-28)

ID	Title	Definition-of-Done	Points
WEB3-5	`POST /prepare_transaction` endpoint	Returns EIP-1559 compatible tx	3
CHAT-1	Store `user_wallet` in `chat_sessions` table	Migration + CRUD tests	3
WS-1	Forward unsigned tx via WebSocket	Flutter receives JSON	3
SEC-1	No private keys logged server-side	Snyk scan pass	2

Sprint Goal: Agent prepares tx → Flutter pops signer.

Sprint 3 – **Wallet Connect & Sign**

ID	Title	Definition-of-Done	Points
FLUT-1	Integrate `web3auth_flutter`	Login with Google → address	5
FLUT-2	`signTransaction()` service	Returns signed hex	3
CHAT-2	Broadcast signed tx & return hash	Hash visible in chat bubble	3

Demo Video: 30-second clip of agent sending 0.001 ETH on Goerli.

Sprint 4 – **NFT Memory Mint**

ID	Title	Definition-of-Done	Points
SC-1	Deploy `AgentMemoryNFT.sol` to testnet	Verified contract + ABI	3
WEB3-6	`POST /mint_memory_nft` tool	Returns unsigned mint tx	3
CHAT-3	Trigger mint on 5-star rating	NFT visible on OpenSea	4
FLUT-3	NFT gallery screen	Grid view + metadata	3

Sprint Goal: Every "great answer" = new NFT.

Sprint 5 – **USDC Payments**

ID	Title	Definition-of-Done	Points
SC-2	Deploy `PaymentEscrow.sol`	Unit tests 100 %	3
PAY-1	`deduct_payment()` MCP tool	Checks USDC balance	3
CHAT-4	Token-gate premium agent	Reject if balance < 1 USDC	3
FLUT-4	Show USDC balance widget	Real-time update	2

Sprint 6 – **Launchpad UI & QA**

ID	Title	Definition-of-Done	Points
FLUT-5	Launchpad create-agent flow	Form → deploy → mint NFT	5
QA-1	End-to-end Cypress tests	10 happy-path scenarios	3
DOC-1	Public docs + README badges	Netlify deploy	2
REL-1	Tag `v4-web3-GA`	GitHub release notes	1

🎟 Sample User Stories (Product Backlog)

As a …	I want …	So that …
DeFi user	my agent to auto-swap tokens	I capture arbitrage without manual clicks
DAO member	the agent votes for me	I never miss governance proposals
Creator	each insight minted as NFT	I monetize my agent’s IP

🔍 Sprint Ceremonies Calendar

Ceremony	Cadence	Owner
Sprint Planning	Monday 09:00 UTC	PM + Team
Daily Stand-up	09:15 UTC (15 min)	Scrum Master
Review / Demo	Friday 16:00 UTC	Dev Team
Retrospective	Friday 16:30 UTC	Scrum Master

📊 Burndown & Velocity

- Story points per sprint: ~20
- Velocity tracked in GitHub Projects “IRIS-Web3” board.
- Burn-down chart auto-generated via GitHub Actions + Plotly.

Copy-paste this board into **GitHub Projects** or **Jira**, assign owners, and start sprinting toward a **live AI × Web3** product.