Digital Mujtahid: Judgment for Builders

Ten principles for making product decisions with evidence, purpose, and conscience

Falah Digital

Digital Mujtahid: Judgment for Builders

Ten principles for making product decisions with evidence, purpose, and conscience.

How to Use This Course

Every builder faces the same recurring moment: a decision arrives faster than the evidence for it. A founder bursts into the sync with a feature already announced. A metric is spiking and someone wants to double down. The codebase is groaning and the roadmap will not wait.

This course treats each of those moments the way a *mujtahid* — a qualified jurist exercising reasoned judgment — treats a question of law: not with a reflex, but with a method. You investigate the problem (*istiqsa'*), weigh the evidence (*istidlal*) by its real strength (*qati'* / definitive vs *zanni* / probable), consult those affected (*shura*), and only then issue a ruling (*hukm*) — bound by stated conditions (*shurut*) and nullifiers (*munkathirat*), and anchored to a higher purpose (*maqsad*).

Each of the ten lessons dramatises one decision through two voices in the same head — the **Product Lead** (frameworks, data, trade-offs) and the **Mujtahid** (purpose, evidence hierarchy, conscience) — then crystallises the teaching into a **Principle (Hukm)**, a three-step **Protocol** you can run this sprint, and a **Muhasaba** (retrospective) question designed to make you honest.

Read one lesson per sitting. Run its Protocol before the next. At the end, use the **Decision Memo Template** to turn any live decision into a founder-grade ruling.

A note on sources: this material is adapted faithfully from the *Digital Mujtahid* manuscript. Two principles (7 and 9) survive only in part in the source; the surviving text is preserved as-is rather than invented around.

Lesson 1. The Problem Statement — What Problem Are We Solving?

*Every build decision begins with a ruling (*hukm*) on the problem itself. Before a line of code, the builder-as-*mujtahid* investigates the problem space (*istiqsa'*), frames it as a Job-to-be-Done, and runs it through the Maqasid Problem Filter. This lesson shows how to refuse a feature request until a validated problem statement exists.*

1. THE SCENARIO

You're the Product Lead at a fintech startup that just closed Series A. Monthly active users are growing, but retention is flat. Your CEO bursts into your weekly sync: "Our users aren't saving enough. Let's build an auto-round-up feature—every purchase, round up to the nearest dollar and stash the difference. It's what Monzo and Acorns do. Ship it by end of quarter."

You open your notebook. The spreadsheet of user interview notes stares back at you. The Mujtahid in you whispers: *Before you write a single line of a PRD, ask: What is the hukm (ruling) on this decision? What is the maqsad (purpose) we are truly serving?* The Product Lead in you adds: *Is this a symptom or a root cause? Is this user pain or business pain? And most importantly—which job are we actually being hired to do?*

---

2. DISCOVERY (ISTIQSA')

PRODUCT_LEAD:
Stop. Don't touch a wireframe. We're not building anything until we understand the problem space. I pull out the **Opportunity Solution Tree** framework. I ask the team: *What outcome do we want?* "Increase savings rate" is not an outcome—it's an output. The real outcome? *Users feel financially secure enough to focus on other parts of life*. That smells like **Hifz al-Mal** (preservation of wealth) and **Hifz al-Aql** (preservation of mind/clarity).

I draw the tree. Root node: *Improve financial well-being*. First-level opportunities: (1) Users don't know where their money goes. (2) Users lack the discipline to save. (3) Users feel anxious about unexpected expenses. The CEO's feature—auto-round-up—maps to opportunity #2. But is #2 the root? I schedule 10 user interviews this week. No solution talk. Just JTBD: *When you think about saving money, what's the job you're trying to get done?* The answer might be: "I want to stop feeling guilty about spending on coffee." That's an emotional job, not a functional one.

MUJTAHID:
Before a fatwa, I conduct **Istiqsa'**—exhaustive investigation of the problem space. The Prophet ﷺ said: "The judge must not judge between two people while he is angry." Anger clouds reasoning. Likewise, the CEO's excitement for a popular feature is a form of *hawa* (desire). I ask: *What is the maqsad al-shar'i (higher objective) of this product?* It is not merely to make users

save—it is to protect their *aql* (mind) from the anxiety of financial insecurity, which is a *daruriyyah* (essential need) under Hifz al-Aql.

I then map the problem using **Maqasid lenses**:
- Is this a *daruriyyah* (essential), *hajiyyah* (needed), or *tahsiniyyah* (embellishment)?
- If users don't save, do they lose their mental clarity (aql)? Or do they just miss an opportunity for optimization?

The CEO's feature assumes the problem is *lack of a mechanism*. But Istiqsa' demands we test that assumption. I ask: *What evidence do we have that users actually want to save?* If the problem is *lack of knowledge* (opportunity #1), then auto-round-up is a *sad al-dhara'i* (blocking the means) that blocks spending—but it may also block the *maslaha* (benefit) of conscious budgeting. I record all assumptions as *zanni* (speculative) until proven.

Key question from Istiqsa': How do we distinguish a symptom from a root cause? A symptom is the surface complaint—"I can't save." A root cause is the underlying *mafsada* (harm) or *maslaha* gap—e.g., "I have no visibility into my spending pattern, so I feel out of control." The Mujtahid asks: *What is the 'illah (effective cause) behind the user's behavior?* Is it ignorance, lack of tools, or emotional friction? The Product Lead asks: *What is the JTBD that the user is hiring our product to do?* Both converge on the same truth: the problem statement must describe the job, not the solution.

---

3. EVIDENCE (ISTIDLAL)

PRODUCT_LEAD:
Now we gather daleel. Two buckets: **Quantitative** and **Qualitative**.

Quantitative: I pull analytics. Our users who attempt to set a savings goal drop off at 70% mid-funnel. That's a signal. But is it *qati* (certain)? No—it's *zanni* (speculative) because the drop-off could be due to UI confusion, not lack of desire. I need cohort analysis: do users who complete setup actually save more? If yes, then the bottleneck is *completion*, not *desire*.

Qualitative: I conduct 10 user interviews. One user says: "I tried rounding up for two weeks, but I felt like I was being punished for buying groceries. I stopped." That's a *symptom*—the feature itself created a negative emotional job (punishment). Another user says: "I want to save automatically, but I also want to know exactly how much I'm saving every month." That reveals a *missing job*: transparency.

I map these onto the Opportunity Solution Tree. The evidence points to opportunity #1 (lack of visibility) being more fundamental than #2 (lack of discipline). I update my hypothesis: *The job is not "make me save" but "show me my financial reality so I can make confident decisions."*

MUJTAHID:
Evidence in Usul al-Fiqh is categorized by strength:
- **Qati al-Thubut** (certain transmission): data that is indisputable—e.g., server logs showing drop-off rates.
- **Zanni al-Thubut** (speculative transmission): user quotes—they might be lying or mistaken.

- **Qati al-Dalala** (certain meaning): the interpretation is obvious—e.g., "I stopped because I felt punished" clearly indicates emotional pain.
- **Zanni al-Dalala** (speculative meaning): a high drop-off rate could mean multiple things (bad UI, low intent, technical failure).

I weigh the evidence: The quantitative drop-off is *qati al-thubut* (the data is fact) but *zanni al-dalala* (the cause is ambiguous). The qualitative quote is *zanni al-thubut* (one user's opinion) but *qati al-dalala* (the emotional job is clear). To strengthen, I apply **Istishab** (presumption of continuity): we presume no problem exists until evidence contradicts. Here, the drop-off contradicts the assumption that users want auto-save. So the default state is: *the current solution is not solving the real problem.*

I also apply **Sad al-Dhara'i** (blocking the means): if we build auto-round-up without solving visibility, we may actually increase anxiety (users feel they lost control). That harms Hifz al-Aql. The evidence does not support moving forward.

Key lesson from evidence: Distinguish anecdote from data. One user's "I felt punished" is an anecdote—but if it resonates with patterns across 8 of 10 users, it becomes *mustanid* (supported evidence). The Product Lead triangulates; the Mujtahid demands *tawatur* (multiple independent chains) before accepting.

---

4. SHURA (CONSULTATION)

PRODUCT_LEAD:
I call a cross-functional sync: Engineering, Design, Data, and the CEO. I share the Opportunity Solution Tree and the evidence. I start with the user quotes—let them feel the pain. Then I ask: *What do we think the root cause is?*

The CEO says: "But auto-round-up works for everyone else." That's an appeal to popularity (*istihsan bi al-'adah*). I counter: "We need to solve *our* users' job, not copy features."

The Data Lead points out: "Our retention curve flattens after 30 days—users who save manually stay longer than those who use auto-features." That's *qati al-thubut* evidence.

Design says: "What if we build a dashboard first? Show spending breakdown before any saving feature."

I record all input. I don't dismiss the CEO—I note his concern as a *maslaha* (speed to market). But I weight it lower because it contradicts user evidence. I then ask: *Who else should we consult?* We need to talk to users who churned. I schedule 5 more interviews.

MUJTAHID:
Shura is not a vote. It is a method of *ijtihad jama'i* (collective reasoning). The Prophet ﷺ consulted the companions even when he knew the answer—to build ownership and gather insights.

I establish *adab al-shura* (etiquette of consultation):
1. **No rank-based weighting.** The CEO's opinion gets the same initial weight as the intern's. True evidence trumps authority.
2. **Record dissent explicitly.** If Engineering says "this is technically hard to roll back," that

becomes a *shart* (condition) for the decision.
3. **Seek those with *khibra* (expertise).** The Data Lead's analysis is stronger than a generalist's intuition. I assign a weight of 3x to domain experts.

The key *mas'ala* (issue) is: *Is the problem we're solving the right one?* The shura reveals consensus that auto-round-up addresses a symptom, not the root. But we need one more round: consult the *ahl al-khibra*## THE PRINCIPLE (HUKM)

HUKM: We will not build any solution — not a single line of code, not a prototype, not an A/B test — until we have a validated problem statement that passes the **Maqasid Problem Filter** and is framed as a **Jobs-to-be-Done**.

DALEEL:
Quantitative data from our analytics shows a 40% drop in daily active users within the first week of new feature launches — not because the features were broken, but because they solved problems nobody had. Qualitative interviews (n=12) revealed users couldn't articulate *what job they were hiring our app to do*; they described tasks, not outcomes. The Shura session with stakeholders confirmed that we have been building from "cool idea" rather than "real struggle." The evidence is **zanni al-dalala** (probable in meaning) but **qati' al-thubut** (certain in occurrence): we are wasting engineering capacity on assumptions.

MAQSAD: Primary: **Hifz al-Aql** (Preservation of Mind) — by forcing ourselves to think clearly about the true problem, we protect the team's cognitive energy from being scattered across low-impact work. Secondary: **Hifz al-Mal** (Preservation of Wealth) — we stop burning budget on features that don't move the North Star metric.

SHURUT (Conditions that must be met before the principle is considered fulfilled): - The problem statement must include a **JTBD functional job** (e.g., "When I read Quran, I want to capture the verse that struck me so I can reflect on it later") — not a feature request. - The problem must pass the **Maqasid Filter**: does solving it preserve Aql, Din, Nafs, Mal, or Nasl? If none, deprioritize. - The problem must be validated with at least **5 user interviews** where the user's *struggle* is observed, not just stated. - The problem must have a **measurable outcome** (e.g., increase in "verse capture rate" by 20%) before any solution sketch is drawn.

MUNKATHIRAT (Nullifiers — conditions that invalidate this principle): - If a stakeholder (including the CEO) bypasses the problem statement and says "just build this feature," the principle is nullified — we must immediately call a Shura to re-establish discipline. - If the team starts wireframing before the problem statement is written in the JTBD format, the principle is broken — halt and rewrite. - If the problem statement changes mid-sprint without a new validation cycle, the principle is violated — rollback to discovery.

THE PROTOCOL

STEP 1: Write the JTBD Problem Card (by Tuesday)

Take the feature request your team is most excited about. Write it on a physical index card. Flip it over. On the back, answer: *What job is the user trying to get done? What is the struggle?*

What progress are they trying to make? If you cannot write a clear functional job in one sentence, do not proceed. This is your **"istidlal al-mas'ala"** (inference of the problem).

STEP 2: Run the Maqasid Filter (Wednesday)
Bring the problem card to a 30-minute standup with the product triad (PM, designer, lead engineer). For each Maqsad (Din, Nafs, Aql, Mal, Nasl), ask: *If we solve this problem, which Maqsad do we serve? How? Give one concrete example.* If the problem serves zero Maqasid, kill it. If it serves one but with negative externalities (e.g., increases user anxiety), apply **Sad al-Dhara'i** (blocking the means to harm) and deprioritize.

STEP 3: Validate with 5 Struggles, Not 5 Opinions (by end of Sprint)
Conduct 5 user interviews. Do not ask "Would you use X?" Ask: *Tell me about the last time you tried to do [the job]. What made it hard? What did you do instead?* Record each struggle as a quote. After the interviews, rewrite the problem statement based on what you observed. If the statement changes, you have validated the *real* problem. If it stays the same, you may have confirmation bias — interview 3 more users.

MUHASABA (RETROSPECTIVE)

What would our product look like if we deleted every feature that was built without a validated problem statement?

Take that list. Burn it in your mind. Now ask: *What is the one feature we would keep — the one that actually solves a real struggle for a real person?* That is your **North Star**. Everything else is decoration. Start there next sprint.

Lesson 2. Continuous Discovery — Evidence as Istidlal

*A product decision is a ruling, and a ruling without evidence (*daleel*) is built on sand. This lesson introduces the Evidence Quadrant — Quantitative, Qualitative, Behavioral, Attitudinal — and the Usul hierarchy of certainty (*qati'* vs *zanni*) you use to decide how much evidence a given decision actually demands.*

MAQSAD: Hifz al-Mal (Preservation of Wealth/Data) **FRAMEWORK:** Evidence Quadrant – Quantitative / Qualitative / Behavioral / Attitudinal

---

1. THE SCENARIO

You're the Product Lead at a fast-growing fintech startup. Your CEO bursts into your weekly sync: "Why are we building this feature? Show me the evidence." You open your notebook. The Mujtahid in you asks: *What is the hukm? What is the maqsad?* The Product Lead in you asks: *What did we discover? What opportunity are we solving?* The CEO isn't asking for a roadmap slide. She's asking for the daleel. And if you can't produce it, you're building on sand. This is a

Hifz al-Mal moment — preserving the company's wealth (time, engineering hours, user trust) from waste. The Habit of Continuous Discovery is your shield.

---

2. DISCOVERY (ISTIQSA')

PRODUCT_LEAD:
Continuous Discovery is not a meeting. It's a weekly cadence of talking to users, mapping opportunities, and testing assumptions before writing a single line of code. You start with the **Opportunity Solution Tree**. Draw the trunk: the desired outcome (e.g., "Increase monthly active savings accounts by 20%"). Then branch into opportunities: *"Users don't trust automated savings rules"*, *"Onboarding takes too long"*, *"No clear feedback loop after deposits"*. Each opportunity is a hypothesis. Then branch into solutions: *"Add manual override for auto-save rules"*, *"Streamline KYC with biometrics"*, *"Push notification after each deposit"*. But before you pick a solution, you must gather evidence. The habit is: every week, talk to 2–3 users about the opportunity space. No sales pitch. Just JTBD interviews: *"What job were you trying to get done when you set up your savings rule?"* Discovery is a habit, not a phase. You do it continuously because the problem space shifts like sand.

MUJTAHID:
Istiqsa' (إستقصاء) in Usul al-Fiqh means thorough investigation — leaving no stone unturned before issuing a hukm. A Mujtahid does not rule on a matter until she has exhausted the sources: Qur'an, Sunnah, Ijma', Qiyas, and then the subsidiary proofs (Istihsan, Maslaha, etc.). She does not leap from a single hadith to a fatwa. She asks: *Is this daleel qati' (definitive) or zanni (probable)? Is the chain authentic? Does it conflict with a stronger daleel?* This is exactly the rigor of Continuous Discovery. The product decision is your hukm (ruling). The maqsad is Hifz al-Mal — preserving the company's wealth and the user's wealth (data, time, money) from waste. Istiqsa' means you do not build a feature based on a single user request or a CEO's gut feeling. You investigate: *What is the real problem? Who faces it? Under what conditions? What have we learned from past features?* The Mujtahid's question at this stage: *What constitutes 'sufficient evidence' for this product decision?* The answer depends on the risk. High-risk decisions (e.g., changing the core transaction flow) require qati' evidence — multiple data sources, triangulated user interviews, A/B test results. Low-risk decisions (e.g., button color) can be zanni — a heuristic or a single test. But the habit of Istiqsa' demands you know the difference.

---

3. EVIDENCE (ISTIDLAL)

PRODUCT_LEAD:
Evidence comes in four flavors — the **Evidence Quadrant**:
- **Quantitative** (what: metrics, dashboards, funnels)
- **Qualitative** (why: interviews, diary studies, usability tests)
- **Behavioral** (what users actually do: logs, clickstream, feature adoption)
- **Attitudinal** (what users say: surveys, NPS, feedback forms)

You need all four to triangulate. A high drop-off in onboarding (quantitative) tells you *where* people leave. Interviews (qualitative) tell you *why* — maybe they're confused by the savings rules. Behavioral data (logs) shows that 70% of users who attempt to set a rule abandon after step 3. Attitudinal data (survey) reveals 60% find the rule builder "too complex." Now you have evidence. But the habit is: you don't wait for a perfect dataset. You gather **minimum viable evidence** — just enough to reduce uncertainty to a level where you can make a decision. If the risk is low, a single interview with a clear pattern may suffice. If the risk is high (e.g., launching a new payment rail), you need a full investigation: user research, competitive analysis, prototype testing, and a staged rollout.

MUJTAHID:
In Istidlal (استدلال), we classify evidence by strength. The hierarchy:
- **Qati' al-Thubut wa Qati' al-Dalala** — Definitive in transmission and meaning (like a mutawatir hadith). In product: a clear, replicable A/B test with statistical significance, or a consistent pattern across 20+ user interviews.
- **Qati' al-Thubut wa Zanni al-Dalala** — Definitive transmission but ambiguous meaning. Example: a metric that is reliable but could be interpreted multiple ways (e.g., "time on page" could mean engagement or confusion).
- **Zanni al-Thubut wa Qati' al-Dalala** — Probable transmission but clear meaning. Example: a single user interview where the user's words are unambiguous, but you need more data to confirm it's not an outlier.
- **Zanni al-Thubut wa Zanni al-Dalala** — Both weak. Example: a gut feeling from the CEO, or a survey with a biased sample.

Your job as product Mujtahid is to weigh the evidence and assign a confidence level. The **maqsad of Hifz al-Mal** means you protect resources from being wasted on low-confidence decisions. You do not build a feature based on zanni evidence alone — unless the cost of delay is higher than the cost of a mistake (Maslaha principle). Signal vs. noise: a single spike in support tickets about a bug is signal. A low NPS score with no qualitative context is noise. You must filter noise by triangulating across quadrants. The **minimum viable evidence** for a principle (a product decision with high impact) is at least two independent sources from different quadrants, with no contradictory evidence. For an MVP experiment, one strong qualitative signal plus a behavioral data point is sufficient.

---

4. SHURA

PRODUCT_LEAD:
Discovery is not a solo sport. You need input from stakeholders: engineering (feasibility), design (usability), data science (metrics), customer support (pain points), and legal (compliance). But Shura is not voting. It's consultation with weight. You present your evidence — the Opportunity Solution Tree, the user quotes, the data — and ask: *What am I missing? What assumptions are we making? What risks do you see?* You record dissent explicitly. If an engineer says "This will break our caching layer," that's a shart (condition) you must address before the hukm. If a

designer says “Users will find this confusing,” that’s a potential munkathir (nullifier) — if proven, the decision must reverse. Shura is not a rubber stamp. It’s a pressure test of your evidence.

MUJTAHID:
Shura (مشاورة) is an obligation for the Mujtahid when the evidence is not definitive. The Prophet ﷺ consulted his companions even when the revelation was clear, to teach the method. The methodology:
1. **Identify the right stakeholders** — those with relevant expertise (not just hierarchy). The engineer who built the legacy code, the customer support agent who hears complaints daily, the finance analyst who knows the unit economics.
2. **Present the evidence neutrally** — do not lead the consultation with your preferred conclusion. The Mujtahid states the facts, the opportunities, the risks.
3. **Weigh the input** — not all voices are equal. The person closest to the user’s pain may have more weight than the VP who hasn’t spoken to a customer in a year. Weight by relevance, not rank.
4. **Record dissenting opinions** — if a respected stakeholder disagrees, note it in your product decision log. That dissent becomes a munkathir (nullifier) if later evidence proves it correct. It also protects you from groupthink.

A Muslim PM should see Shura as a sunnah that protects Hifz al-Mal — the collective wisdom of the team prevents costly mistakes. The output of Shura is not consensus; it’s a refined set of shurut (conditions) and munkathirat (nullifiers) that you will carry into the principle.

HUKM: We will institutionalise a weekly Continuous Discovery habit that systematically collects evidence across all four quadrants of the Evidence Quadrant (Quantitative, Qualitative, Behavioral, Attitudinal) before committing any engineering resources to a new feature or experiment.

DALEEL: Analysis of our last three sprints shows 67% of shipped features failed to move the North Star metric because decisions were based on attitudinal data alone (customer “I want X” interviews) without behavioral validation. The Evidence Quadrant framework, when applied rigorously, reduces waste (Hifz al-Mal) by grounding decisions in triangulated signals. Our own A/B test history confirms that features validated with ≥3 quadrants have a 4× higher success rate.

MAQSAD: Primary: *Hifz al-Mal* (Preservation of Wealth/Data) – avoiding wasteful engineering spend and protecting the integrity of our data pipeline. Secondary: *Hifz al-Aql* (Preservation of Intellect) – ensuring decisions are made with sound reasoning, not guesswork.

SHURUT: * Evidence must be refreshed within the last 14 days before any sprint commitment. * At least two quadrants must agree before a decision is taken to build. * Behavioral data (actual user interaction logs) takes precedence over attitudinal data (stated preferences) in case of conflict. * The Evidence Quadrant board must be visible to the entire team and updated weekly.

MUNKATHIRAT: * Skipping the weekly evidence review without a documented emergency. * Making a build decision based on a single quadrant (e.g., only quantitative metrics or only a

CEO's opinion). * Ignoring disconfirming evidence that emerges after a decision is made – the principle is invalidated if the team refuses to revisit the evidence.

THE PROTOCOL

STEP 1: Every Monday morning, block 90 minutes for the Continuous Discovery huddle. The PM brings three pieces of evidence: one quantitative (e.g., funnel conversion), one qualitative (e.g., user interview quote), and one behavioral (e.g., session recording clip). No slide decks – just raw evidence on the Evidence Quadrant board.

STEP 2: For any feature being considered for the next sprint, the team must map it to at least two quadrants within that same week. If only one quadrant has evidence, the feature is deferred to the next cycle. Use the Opportunity Solution Tree to trace the evidence back to a specific opportunity.

STEP 3: Every Friday, run a 30-minute "Evidence Audit" where the team asks: "What did we learn this week that disconfirms our current assumptions?" If any Munkathirat condition is triggered, the build is paused and the Hukm is re-evaluated before Monday.

MUHASABA (RETROSPECTIVE)

Where in our last sprint did we ship something based on a single quadrant, and what would it have cost us if we had been wrong – both in wasted engineering hours (Hifz al-Mal) and in misleading data that now pollutes our model (Hifz al-Aql)?

Lesson 3. Stakeholder Consultation — Shura as Discovery

*Consultation (*shura*) is not a vote and not performative noise; it is a method of evidence-gathering that preserves the dignity (*Hifz al-Karama*) of everyone affected. This lesson maps stakeholders into the Shura Quadrant — Users, Team, Business, Regulators — and shows how to weigh conflicting voices by evidence and purpose, not by rank.*

Maqasid: Hifz al-Karama (Preservation of Dignity) → Stakeholder Voice
Framework: Shura Quadrant — Users / Team / Business / Regulators

---

1. THE SCENARIO

You're the Product Lead at a growing fintech startup. Your CEO walks into your desk, coffee in hand, and asks: "Why are we building this feature? And whose voice are we listening to?" You open your notebook. The Mujtahid in you asks: *"What is the hukm of consulting stakeholders?*

What is the maqsad behind their input? And what happens when their voices conflict — whose daleel carries weight?" The CEO isn't waiting for theory. She wants a decision by end of week. You need a framework that turns consultation into discovery, not performative noise.

---

2. DISCOVERY (ISTIQSA')

PRODUCT_LEAD:
Continuous Discovery starts with a clear Opportunity Solution Tree. Your CEO's question is a gift — it forces you to map the problem space. Who are your stakeholders? Users, engineers, compliance, finance, the CEO herself. Each sits in a different branch of the tree. Your job is not to collect opinions but to uncover the underlying *jobs to be done* for each stakeholder.

- **Users:** What outcome do they want from this feature? (e.g., faster checkout, lower fees)
- **Team (engineers, design):** What technical or process constraints do they see? (e.g., legacy API, sprint capacity)
- **Business (CEO, finance):** What metric moves? (e.g., conversion rate, LTV)
- **Regulators (legal, compliance):** What risk must be mitigated? (e.g., KYC, data privacy)

Run a **Shura Quadrant** mapping exercise this week. Draw four boxes. Populate them with the *primary job* each stakeholder is trying to get done *through you*. Then ask: *Which box is empty?* That's where discovery is needed.

MUJTAHID:
In Usul al-Fiqh, *Istiqsa'* (comprehensive investigation) is the systematic search for all relevant evidence before issuing a ruling. Shura is not a brainstorming session — it's a *method of evidence gathering*. The Prophet ﷺ consulted his Companions on matters where revelation was silent, not to poll feelings but to surface *valid maqasid* (purposes) and *masalih* (benefits).

Your fintech feature request is a *mas'ala* (question). The stakeholders are *ahl al-shura* — people with relevant knowledge, not equal weight. The Mujtahid's question: *What is the maqsad behind each stakeholder's voice? Is it Hifz al-Mal (preservation of wealth)? Hifz al-Nafs (safety)? Hifz al-Aql (sound judgment)?*

- **Users:** Their voice protects *Hifz al-Mal* (fair pricing, no exploitation) and *Hifz al-Karama* (they are not just data points).
- **Team:** Their voice protects *Hifz al-Aql* (technical soundness, avoid harm from rushed code).
- **Business:** Their voice protects *Hifz al-Mal* (sustainability, growth) but can become *gharar* (uncertainty) if over-prioritized.
- **Regulators:** Their voice protects *Hifz al-Din* (contractual validity, riba avoidance) and *Hifz al-Nafs* (fraud prevention).

The maqsad of Shura is not consensus — it is *istiqsa' of masalih and mafasid*. You must discover whose input is *dareuri* (essential) vs *hajiyyat* (needed) vs *tahsiniyyat* (nice-to-have). The empty quadrant is a *mafsada* waiting to happen.

---

3. EVIDENCE (ISTIDLAL)

PRODUCT_LEAD:
Evidence comes in two flavors: quantitative (metrics, A/B tests, analytics) and qualitative (interviews, observation, support tickets). Your CEO wants a decision — but the feature request is likely based on a single data point (e.g., "our competitor has it"). That's *zanni* (speculative) evidence at best.

Build your evidence stack:

- **Quantitative:** Funnel drop-off rates, feature usage heatmaps, survey NPS segmented by stakeholder group. *What does the data say about the problem's prevalence?*
- **Qualitative:** 5 user interviews this week, 2 stakeholder shadow sessions (e.g., sit with compliance during a manual review). *What story does the data not tell?*

When stakeholders conflict (e.g., users want speed, compliance wants more checks), weight evidence by *frequency, severity, and reversibility*. A 2% conversion loss from adding a step is *zanni* until you validate it with a cheap experiment. A regulatory fine of $10M is *qati'* (definitive) — it must be addressed first.

MUJTAHID:
Daleel in Usul is ranked: *Qati' al-Thubut wa Qati' al-Dalala* (definitive source and meaning) vs *Zanni* in either. In Shura, stakeholder input is rarely *qati'*. It is *zanni* evidence that must be weighed.

- **Qati' evidence:** Explicit regulatory requirements, binding contracts, clear user harm (e.g., bug causes financial loss). These override everything.
- **Zanni evidence:** CEO's strategic hunch, engineer's "I think this is easier," user's stated preference (which may differ from behavior).

The Mujtahid uses **tarjih** (preponderance) rules:

1. **Who holds the burden of proof?** The stakeholder advocating a change must bring *daleel*. If they say "competitor does it," that's *zanni* — insufficient.
2. **What is the maqsad?** If the feature protects *Hifz al-Din* (e.g., ensuring riba-free transactions), it takes precedence over *Hifz al-Mal* (revenue).
3. **Sad al-Dhara'i* (blocking the means to harm):** If a feature opens a path to *mafsada* (e.g., easier fraud), it is avoided even if users request it.
4. **Istishab (presumption of continuity):** Assume the current state is valid until evidence proves otherwise. Don't build a feature just because someone asked — the default is *no change*.

Use a simple **Tarjih Matrix**: For each stakeholder input, assign a score (1-3) for *Qati'/Zanni*, *Maqsad weight* (Daruri=3, Hajji=2, Tahsini=1), and *Mafsada risk* (high=3, medium=2, low=1). The highest total wins — but only if the maqsad is preserved.

---

4. SHURA

PRODUCT_LEAD:
Shura is not a meeting. It is a structured discovery ritual. You don't just "ask stakeholders what they want" — you build a **Shura Cadence**:

- **Weekly 30-min sync** with each quadrant (Users, Team, Business, Regulators). Use a consistent format: *"What problem are we solving? What evidence do you have? What outcome do you need?"*
- **Monthly Shura Council** — one hour, all four quadrants present. Each presents their top opportunity (backed by evidence). The goal is not to vote but to *surface trade-offs*.
- **Record dissent explicitly.** When a stakeholder disagrees, write their dissent and the *daleel* they offered. This is *tadwin al-shura* — documenting consultation. It protects *Hifz al-Karama*: their voice was heard, weighted, and preserved.

MUJTAHID:
The Mujtahid consults by *tamyiz* (discernment), not democracy. The Prophet ﷺ consulted Badr strategy with the Companions, but the final decision was his. Shura is advisory, not binding — but the advisor's dignity is sacred.

How to consult as a Mujtahid PM:

1. **Start with the maqsad.** Open the Shura by stating: *"We are here to protect $MAQSAD for $STAKEHOLDER. What evidence do you have that $FEATURE serves or harms this?"*
2. **Weight by expertise, not hierarchy.** The engineer's technical feasibility is *zanni* but high weight in that domain. The CEO's strategic vision is *zanni* but high weight in market context. The user's pain is *zanni* but weight increases with frequency.
3. **Use the Shura Quadrant to identify gaps.** If only Business speaks, ask: *"Where is the user's voice?"* Silence is also a signal — it may indicate *hifz al-Karama* is being violated (their input wasn't solicited).
4. **End with a ruling, not a poll.** After gathering all input, you (the Product Lead / Mujtahid) issue a *hukm*: "We will proceed with $DECISION, based on $DALEEL, with $SHURUT (conditions), and $MUNKATHIRAT (rollback triggers)." Record the reasoning. This is *tadwin al-ijtihad*.

Shura is the discovery engine. When done right, it transforms stakeholder consultation from a political chore into a rigorous, dignity-preserving investigation of what is *khayr* (good) for the product and its people.

HUKM: We will build a recurring, structured Shura Council for every major product decision, with rotating representation from Users, Team, Business, and Regulators, and we will publish a summary of input and outcomes.

DALEEL: Discovery interviews across 12 stakeholder interviews revealed that 8 felt their voice was either ignored or collected too late. Quantitative survey (n=50) showed 74% of stakeholders would increase trust with a formal consultation process. Maqasid analysis confirms that Hifz al-Karama (dignity) is violated when stakeholders are informed after decisions are locked. Qiyas: the Prophet ﷺ consulted even in minor matters (Khandaq, changing strategy based on Salman al-Farsi's input). The daleel is strong: recurring Shura preserves dignity AND improves decision quality.

MAQSAD: Hifz al-Karama – the preservation of stakeholder dignity through genuine voice, which is a necessary condition for trust, alignment, and sustained collaboration. Secondary Maqasid: Hifz al-Mal (business outcomes improve when blind spots are surfaced early).

SHURUT: - The Shura Council must include at least one representative from each quadrant (Users, Team, Business, Regulators) per decision. - A written "Shura Brief" (problem statement, constraints, options) must be distributed 48 hours before the session. - The Product Manager must explain how each piece of input was used or why it was not used, and publish this in a public log. - Rollback trigger: If any quadrant misses two consecutive sessions, the decision is paused until they are represented.

MUNKATHIRAT: - When stakeholder input is never referenced in the final decision document (nullifies dignity). - When sessions become monologues from the product team (no genuine listening). - When the same individuals dominate every session without rotation (entrenchment, not shura).

THE PROTOCOL

STEP 1: This sprint, identify the next major product decision (e.g., feature prioritization, pricing change, or sunsetting a module). Draft a one-page Shura Brief with the problem, three options, and one clear question: "What would you do if you were me?" Share it with at least one person from each quadrant by Wednesday.

STEP 2: Hold a 45-minute Shura session on Thursday. Start with the brief, then each quadrant speaks uninterrupted for 5 minutes. Take verbatim notes. End with: "I will share my decision and how your input shaped it by Monday."

STEP 3: By Monday, publish the decision log in a visible place (Slack, wiki, or product board). Include: what we decided, which inputs changed our thinking, which inputs we deprioritized and why. Then schedule the next Shura session two weeks out.

MUHASABA (RETROSPECTIVE)

When did you last change a product decision because a stakeholder disagreed with you? If you can't remember, your Shura is performative. Dignity is not about being heard—it's about being able to change the outcome. The real test of Hifz al-Karama is whether your product roadmap has scars from stakeholder voice.

Lesson 4. The Product Decision — The Principle as Definition of Done

The hukm* itself is the Definition of Done — not a checklist, not a ship date. This lesson introduces the Decision Quadrant (Reversible vs Irreversible, High vs Low stakes) and shows how reversibility sets your evidence threshold: reversible decisions tolerate *zanni* evidence; irreversible ones demand *qati'*. The Protocol below is worked through a concrete Shariah-reviewed launch example.*

Maqsad: Hifz al-Din (Preservation of Purpose)
Framework: Decision Quadrant — Reversible vs Irreversible / High Stakes vs Low Stakes / Data-Rich vs Data-Poor
Core Insight: The *hukm* itself is the Definition of Done. Not a checklist. Not a ship date. The principle.

---

1. THE SCENARIO

You're the Product Lead at a growing fintech startup. Your CEO storms into your weekly sync. "We're shipping the instant-payout feature next sprint. Marketing has already announced it. Engineering says two weeks. Go."

You open your notebook. The Mujtahid in you asks: *What is the hukm? What is the maqsad?* The Product Lead in you asks: *Is this reversible? What's the evidence?*

Your CEO expects a simple "yes." But you know: a feature launched without a principled Definition of Done is a fatwa issued without daleel. It binds the team, the user, and the business — sometimes irreversibly.

---

2. DISCOVERY (ISTIQSA')

PRODUCT_LEAD: Let's start with the problem. The CEO says "instant payouts." That's a solution, not a job. What's the Job to Be Done? "I need my money now because my rent is due tomorrow and I can't wait three business days." That's a JTBD: *Help me access my earned wages immediately when I face an urgent cash need.*

Draw an Opportunity Solution Tree. Root: "Users abandon our platform because payout speed is slower than competitors." First branch: "Reduce payout delay." Second branch: "Increase trust

in timing." Third branch: "Provide emergency liquidity." The CEO jumped to a solution (instant payouts) without exploring the tree. We need to climb back up.

MUJTAHID: *Istiqsa'* means exhaustive investigation of the problem space. Before we issue a *hukm*, we must determine the *maqsad* — the intended purpose. What is the *maqsad* of this feature? *Hifz al-Mal* (preservation of wealth)? Yes — faster access to money. But also *Hifz al-Nafs* (preservation of life) — if a user can't pay rent, that's a threat to shelter and stability. And *Hifz al-Din* — the *din* of the product itself: its core purpose. What is the *din* of this fintech? "Help users manage their finances with dignity and speed." Instant payouts preserves that purpose — if done right.

Now classify the decision type. Use the **Decision Quadrant**:

	Reversible	Irreversible
High Stakes	Type 2 (reversible, high stakes) — e.g., a pricing experiment you can roll back	Type 1 (irreversible, high stakes) — e.g., changing the core payout architecture
Low Stakes	Type 4 (reversible, low stakes) — e.g., button color	Type 3 (irreversible, low stakes) — rare, but e.g., deleting old data

Instant payouts is **Type 1** if it changes the underlying ledger system — irreversible because rolling back would break accounting records. It's **Type 2** if it's a front-end flow that can be toggled off. The CEO wants Type 2 speed but the engineering team suspects Type 1 complexity. We must discover which quadrant we're in *before* we decide.

PRODUCT_LEAD: So discovery isn't about "should we build instant payouts?" It's about understanding the decision type. If it's Type 1 (irreversible, high stakes), we need heavy evidence. If Type 2, we can experiment. The Definition of Done for discovery is: *We know which quadrant this decision lives in.*

MUJTAHID: Correct. In Usul al-Fiqh, the *hukm* changes based on the *shart* (condition). The *shart* here is reversibility. A reversible decision allows a *zanni* (speculative) ruling. An irreversible decision demands *qat'i* (certain) evidence. The *maqsad* (preservation of purpose) guides the threshold. If the feature threatens the product's core *din* — e.g., by introducing fraud risk — then *sad al-dhara'i'* (blocking the means to harm) applies even before we have full evidence.

---

3. EVIDENCE (ISTIDLAL)

PRODUCT_LEAD: Now we gather daleel. Quantitative: What percentage of users request instant payouts? What is the current payout speed? What is the churn rate for users who wait 3 days? Data from analytics: 12% of users have complained about payout speed. 30% of those churn within 30 days. That's a signal.

Qualitative: User interviews. "I needed the money for an emergency. I switched to a competitor that offered same-day." That's a JTBD insight. Also: "I'm worried about fees — if instant payouts

cost me $5, I'd rather wait." So the evidence is mixed — urgency exists, but price sensitivity is high.

MUJTAHID: In Usul, we classify evidence by *wurud* (source) and *dalala* (indication). Quantitative data (analytics) is *zanni al-thubut* (speculative in origin) — it depends on instrumentation accuracy. User interviews are *zanni al-dalala* (speculative in meaning) — one user's story doesn't prove the whole population. Both are *zanni*.

For a Type 2 (reversible) decision, *zanni* evidence suffices. But if this is Type 1 (irreversible, high stakes), we need *qat'i* evidence. What would be *qat'i*? A controlled experiment with statistical significance. Or a regulatory requirement (text of law is *qat'i al-thubut*). Or a clear *maslaha* (public benefit) that outweighs harm — but that's a *ijtihad* judgment, not raw data.

Handle uncertainty: The Mujtahid applies *istishab* — presumption of continuity. The current state (3-day payout) is presumed valid until evidence proves otherwise. The burden of proof is on the change. So we ask: "Do we have enough evidence to shift from the default?" The Product Lead asks: "Is the evidence strong enough to justify the risk of reversal?"

PRODUCT_LEAD: That's the same question. The evidence threshold depends on the quadrant. For a reversible experiment (Type 2), we need just enough evidence to justify the experiment — *zanni* is fine. For an irreversible architecture change (Type 1), we need *qat'i* evidence — multiple data sources, consistent user feedback, and a clear business case.

Current evidence: 12% complaint rate, 30% churn among complainers. That's a strong *zanni* signal. But we don't know if instant payouts will actually reduce churn — correlation ≠ causation. We need to test. So the evidence points to Type 2 decision: run a controlled experiment with a subset of users.

MUJTAHID: Agreed. The evidence does not reach *qat'i* for an irreversible change. Therefore, the *hukm* should be: proceed with a reversible experiment, with clear *shurut* (conditions) and *munkathirat* (nullifiers). The *maqsad* of *Hifz al-Din* (preserving the product's purpose) requires that we do not harm the core experience. If the experiment introduces fraud, we must block it (*sad al-dhara'i'*). The Definition of Done for evidence gathering: *We have enough daleel to classify the decision quadrant and set the evidence threshold.*

---

4. SHURA

PRODUCT_LEAD: Time to consult. I schedule a 30-minute decision sync with the CEO, Engineering Lead, and Head of Risk. I bring the Opportunity Solution Tree and the Decision Quadrant. I present the evidence: "We're in Type 2 territory — reversible, high stakes. Here's the data. Here's the user story. Here's the risk of fraud. I recommend an experiment on 10% of users for two weeks."

The CEO pushes back: "Marketing already announced. We can't do a partial rollout." The Engineering Lead says: "We can build a toggle, but the ledger change is one-way if we commit." The Head of Risk says: "Fraud risk is high — we need to cap payouts at $200 per day."

MUJTAHID: Shura is not a vote. It is a structured consultation where each voice is weighted by *khibra* (expertise) and *taqwa* (responsibility). The Mujtahid listens to all, but the final *hukm* is based on evidence and *maqsad*, not consensus.

Let's weight: - CEO: Authority, but also bias toward speed. Weight: medium on evidence, high on strategy. - Engineering Lead: Technical feasibility. Weight: high on reversibility, low on user need. - Head of Risk: Expertise on harm prevention. Weight: high on *sad al-dhara'i'*. - Product Lead (you): Synthesis of evidence and *maqsad*. Weight: highest on the *hukm*.

The dissent here is the CEO's objection to partial rollout. That dissent must be recorded. The Mujtahid asks: "Can we satisfy the CEO's need for a full launch while keeping reversibility?" Answer: Yes — launch to 100% but with a kill switch (toggle) and strict *shurut* (max payout, KYC check). That makes it reversible in practice.

PRODUCT_LEAD: So the consultation produces a refined *shart*: full rollout, but with a toggle and risk controls. The Definition of Done for shura: *We have documented all stakeholder input, weighted it by expertise, and recorded dissent. The final decision is not a compromise — it's a principled ruling based on evidence and maqsad.*

MUJTAHID: Exactly. The *shura* itself is part of the *ijtihad*. It doesn't replace the *hukm* — it informs it. The *hukm* will be: "Ship instant payouts as a reversible experiment with clear nullifiers." That is the Definition of Done. Not a ship date. Not a feature list. A principle.

10% of users using workarounds** (e.g., converting to fiat immediately to avoid holding crypto), the experiment is terminated because it indicates the feature does not serve its intended purpose.

THE PROTOCOL

STEP 1: THIS SPRINT – SECURE THE FATWA

Write a one-page executive summary of the crypto-trading proposal, including the exact token mechanics, smart contract details, and user flow. Submit it to the Shariah board with a deadline of 10 business days. Prepare a sign-off document that explicitly states the permissible conditions (e.g., no leverage, no interest, instant settlement). If the board demands changes, iterate within the sprint.

STEP 2: BUILD THE REVERSIBLE INFRASTRUCTURE

Engineers implement a feature flag toggling the entire crypto module. Write a rollback script that can reverse any pending transactions and restore the previous app state within 60 minutes. Document the rollback procedure and assign an on-call engineer for the beta period. Test the flag and script in staging with simulated transactions.

STEP 3: LAUNCH THE BETA AND MONITOR DAILY

Invite the first 100 users from a waitlist (pre-qualified for high trust). Send them a one-time consent screen with a summary of the Shariah board's ruling and a "I understand the risks" checkbox. Monitor daily: log every transaction, flag any that deviate from the approved mechan-

ics, and run a daily compliance check against the Shurut. At the end of two weeks, hold a retrospective before deciding on broader rollout.

MUHASABA (RETROSPECTIVE)

What single assumption about our users' religious commitment did we make that, if wrong, would invalidate our entire decision?

If the assumption is that users will accurately self-report when they feel a transaction is impermissible, but in reality they stay silent out of fear of missing out, then our "zero complaints" metric becomes meaningless. We would ship a feature that violates Hifz al-Din while believing it is safe. The uncomfortable question: Do we truly know our users' threshold for religious compromise, or are we relying on a proxy that can easily break? Next sprint, add a post-transaction satisfaction survey that explicitly asks: "Did this transaction feel in line with your Islamic values?" Use the answer to calibrate our Shurut, not just as a checkmark.

Lesson 5. MVP as Minimum Viable Principle

An MVP is not just the smallest shippable scope — it is a minimum viable principle*: a conditional decision balancing risk, value, and compliance across the MVP Quadrant (Viable / Valuable / Usable / Feasible). The watch-out this lesson surfaces is letting "Feasible" (what is easy to build) silently override "Viable" (what is right to build).*

Maqsad: Hifz al-Mal (Preservation of Wealth)
Framework: MVP Quadrant — Viable / Valuable / Usable / Feasible
Principle: MVP = Minimum Viable Principle

---

1. THE SCENARIO

You're the Product Lead at a growing fintech startup. The team has been sprinting for six weeks on a new investment feature — fractional shares in halal stocks. The CEO walks into your stand-up: "Why are we building this feature? What's the smallest thing we can ship to learn?" You open your notebook. The Product Lead in you starts sketching an MVP quadrant. But the Mujtahid in you asks a deeper question: *What is the hukm of this feature? What is the maqsad?* You realise that an MVP isn't just about scope — it's about defining a **minimum viable principle**: a conditional decision that balances risk, value, and Islamic compliance. You need to discover, evidence, and consult before you ship.

---

2. DISCOVERY (ISTIQSA')

PRODUCT_LEAD:
You pull out your Opportunity Solution Tree. The CEO's question is an opportunity: "Fractional

shares unlock wealth-building for low-income users." The desired outcome: increase monthly active investors by 20% among users with

3 on ethics/confusion, kill feature. Every Monday morning, the PM runs a 10-minute "Muhāsaba check" — compare actual quadrant performance vs. predicted. If any quadrant drops below threshold, initiate rollback immediately.

MUHĀSABA (RETROSPECTIVE)

Where did we let "Feasible" override "Viable" this sprint, and what wealth did that cost us?
Force yourself to name one feature that was easy to build but ethically uncertain. Write down the exact dollar amount of engineering hours spent, plus the hidden cost of user trust lost. If you can't find a single example, you haven't been honest. The MVP Quadrant is not a rubber stamp — it's a mirror. Look until you see the waste you chose to ignore. Then adjust your Shurūṭ for next sprint.

Lesson 6. Technical Debt — Israf vs Investment

*Not all debt is waste (*israf*). This lesson separates Prudent, Reckless, Deliberate, and Inadvertent debt using the Debt Quadrant, measures debt's true "interest rate" (extra hours each new feature costs), and gives you a Debt Ledger and Debt Covenant to stop calling waste an "investment."*

THE SCENARIO

You're the Product Lead at a growing fintech startup. Your CEO bursts into your weekly sync: "The team is screaming. The frontend is a mess. We need a six-month refactor. But our investors want the next feature in two weeks." You open your notebook. The Mujtahid in you asks: "What is the hukm of this debt? What is the maqsad of the codebase?" Your engineers are burning out. Your users are experiencing bugs. The CEO smells opportunity cost. You smell israf. You need a ruling — not a roadmap.

DISCOVERY (ISTIQSA')

PRODUCT_LEAD: Let's map the problem space. Draw an Opportunity Solution Tree. At the root: "Our codebase slows down delivery." The opportunities: (1) Engineers spend 40% of sprint time fixing old bugs. (2) New features take 3x longer to ship than projected. (3) Onboarding crashes on legacy browsers. The solutions on the table: full refactor, incremental cleanup, freeze all features. But which opportunity do we actually solve? Run a JTBD interview: "When you push code, what job are you hiring the codebase to do?" Answer: "Get to production fast without

breaking things." Current codebase is failing that job. The real opportunity is not "refactor everything" — it's "reduce the friction that makes every feature cost double."

MUJTAHID: Istiqsa' begins with the maqasid. Hifz al-Mal — preservation of wealth. Technical debt is wealth stored in code. If that wealth decays (through bad architecture, duplication, untested paths), it becomes israf. Allah says: *"Indeed, the wasteful are brothers of the devils"* (Quran 17:27). Israf is spending without benefit or spending more than needed. But in Usul, not all debt is israf. Debt can be a means (wasila) to a greater maslaha. So first we define the problem space: What kind of technical debt are we facing? The Debt Quadrant: **Prudent Debt** (taken knowingly, with a plan to repay, for urgent feature that grows revenue) vs **Reckless Debt** (taken without plan, ignoring cost). **Deliberate Debt** (intentional, temporary shortcut with a ticket) vs **Inadvertent Debt** (accidental, from ignorance or lack of process). The israf is in reckless and inadvertent debt — they waste the mal of the company (engineering time, user trust). The maqsad of discovering the debt type is to know whether the current situation is a *darura* (necessity) or a *tahsin* (luxury refactor). The hukm of refactoring depends on the maqsad: are we preserving wealth or just polishing?

EVIDENCE (ISTIDLAL)

PRODUCT_LEAD: Quantitative daleel first. Pull these metrics: Cycle time per feature. Bug reopen rate. Deployment frequency. Code churn (lines changed per feature). Measure the "interest rate" on the debt: How many hours per sprint are burned on unplanned work (bugs, rework)? If that number exceeds 20% of capacity, the debt is compounding. Qualitative daleel: Interview three engineers. Ask: "What's one decision you made last month that you regret because you didn't have time to do it right?" Then ask: "What would you fix first if you had one week?" Also interview one user: "Have you noticed anything getting slower or more broken over the last three months?" Triangulate. If engineers say "we need to rewrite everything" but users say "it works fine," that's a signal. If users are complaining, that's a different daleel. Evidence hierarchy: user behavior (qati' al-dalala — clear indicator) trumps engineer opinion (zanni — probabilistic). But engineer opinion is qati' al-thubut (certain source) about the code's internal state.

MUJTAHID: In Istidlal, we rank evidence by strength. The highest daleel is the *maqsad* itself: Does this debt harm Hifz al-Mal? If the debt causes customer churn (loss of wealth) or blocks a revenue-generating feature (loss of opportunity), then it is israf. Next daleel: *qiyas* — analogize technical debt to a loan with riba. The "interest" on technical debt is the extra time each new feature costs. If the interest rate (extra cost per feature) is higher than the benefit of shipping fast, the debt becomes haram. But if the interest is low (e.g., a 5% overhead) and the feature brings 50% revenue growth, the debt is a valid *istihsan* (juristic preference for necessity). Measure the "interest" using the Product Lead's metrics: extra hours per feature / baseline hours. That ratio is your *riba al-fadl* (excess). If it exceeds 30%, the debt is reckless. Also apply *sad al-dhara'i* (blocking the means to harm): If the team continues accumulating debt without a repayment plan, that path leads to israf. Block it now. But if the debt is deliberate and tracked (e.g., a ticket in the backlog with a deadline), it is permissible as a calculated risk. Write down the

evidence in a table: Type of debt (prudent/reckless/deliberate/inadvertent), interest rate, maqsad harm, and ruling (halal/makruh/haram based on degree of israf).

SHURA

PRODUCT_LEAD: Call a cross-functional sync. Invite: CEO, CTO, lead engineer, one customer support rep, one user (volunteer). Agenda: "We have a codebase problem. Show the evidence above. Now ask each stakeholder: What is the outcome you care about most?" CEO: "Speed to market." CTO: "Quality." Engineer: "Less firefighting." Support rep: "Fewer user complaints." User: "Stability." Now use the Shura methodology: list all priorities, find the intersection. The intersection is: "We need to ship features without breaking things." That's the maqsad. Record dissent: if the CTO insists on a full rewrite, note that as a minority opinion but do not discard it. Shura is not voting; it's weighing evidence and seeking the strongest argument (tarjih). The Product Lead's role: synthesize. Output: a prioritized list of debt items (not a scope of work) ranked by "harm to Hifz al-Mal." The shura decides: "We will pay down only the debt that is directly causing user-facing bugs or blocking a high-value feature. Everything else remains tracked but untouched."

MUJTAHID: Shura in Usul is obligatory for those in authority (Quran 42:38). But the weight of each opinion depends on the *adalah* (trustworthiness) and *khibra* (expertise) of the speaker. The engineer has khibra on code cost. The CEO has khibra on business wealth. The user has khibra on experience. Do not give equal weight. The Mujtahid (you) must weigh: The engineer's claim of "200 hours of debt" is a zanni estimate — it could be exaggerated. The user's report of "app crashes every time I pay" is qati' (certain) in effect. So the shura ruling leans toward fixing crash-causing debt first. Record all opinions, but the hukm is derived from the strongest daleel, not from consensus. Document the shura in a simple table: Stakeholder | Input | Weight (1-5 based on expertise and certainty) | Impact on Ruling. Then write a brief *ta'liq* (annotation) explaining why one view was preferred. This is your shura record — a living artifact for future retrospection.

1. for two consecutive sprints, the entire team halts feature work for one sprint to repay.

- If a single "quick fix" introduces a production incident, the debt is immediately reclassified as Reckless and must be fixed within 48 hours.
- If the team cannot articulate the *maslaha* (expected outcome) for a deliberate debt, the debt is automatically deemed Inadvertent and treated as *Israf*.

THE PROTOCOL

STEP 1: Audit Your Debt Quadrant (This Week)

Take 2 hours as a team. Open your bug tracker and code review history. For each recent shortcut, answer: "Did we plan it? Do we have a repayment date?" Assign it to one of the four quadrants. Create a single "Debt Ledger" document (Google Sheet or Notion) with columns: Type,

Description, Date Incurred, Payoff Date, Interest Cost (estimated hours lost per month). *PRODUCT_LEAD:* Do this during your Tuesday tech review. *MUJTAHID:* This is *istiqsa'* (exhaustive examination) of your *Mal*—do not skip the "why" behind each entry.

STEP 2: Create a Debt Repayment Board (This Sprint)
In your project management tool, add a "Debt Repayment" swimlane. Move all Reckless and Inadvertent items into it. Assign each item a story point value (based on effort to fix) and a DRI (directly responsible individual). Set a sprint goal: "Reduce Reckless debt by 30% (or at least fix the top 3 items)." *MUJTAHID:* This is *tazkiyah* (purification) of your codebase—treat it as a non-negotiable worship of trust (*amanah*).

STEP 3: Define a "Debt Covenant" for Future Sprints (This Sprint Retro)
Write a 1-page team contract:
- "We will never merge a PR without logging its debt impact in the ledger."
- "We will never approve a deliberate debt without a written payoff sprint ID."
- "We will review the ledger every retro and escalate any overdue item."
PRODUCT_LEAD: Make this part of your Definition of Done. *MUJTAHID:* This is your *shart* (condition) that prevents *israf* from recurring—treat it like a legal stipulation in a contract.

MUHASABA (RETROSPECTIVE)

One piercing question for your next retro:
"Where are we knowingly calling Israf* an 'investment' because admitting it's waste would force us to change our roadmap?"*

PRODUCT_LEAD: Look at the last three features you shipped. How many shortcuts did you tell yourself were "necessary to hit the deadline"? Now ask: Did that deadline actually deliver better outcomes, or did you just shift the cost to next quarter?
MUJTAHID: The *mujtahid* knows that *niyyah* (intention) alone does not justify *israf*. If the shortcut wasted more time than it saved, it was *haram* even if you meant well. Measure the actual interest paid (hours lost to bugs, refactoring, onboarding friction). Compare it to the time you "saved." If the debt's interest exceeds its principal, you have committed *israf*—and the only repentance is repayment.

Lesson 7. Pricing & Fairness — Ghabn vs Qima al-Mithl

A just price is not the highest the market will bear — it is the price that reflects genuine worth while preserving the relationship. This lesson carries the pricing decision through Shura with finance, engineering, and an Islamic-finance advisor, testing the model against ghabn* (exploitative gain) and *qima al-mithl* (fair market value). (The source manuscript for this principle is partial; the surviving consultation and retrospective are preserved faithfully below.)*

5% is acceptable.

1. **Islamic finance advisor** (or a scholar versed in muamalat). Present your evidence. Ask: 'Does this pricing model fall under *ghabn*? What is the *qima al-mithl* for a digital product? Is profit-sharing (mudaraba) a better model for our premium tier?' Document their answer as a *fatwa*-like guidance. This becomes your *daleel* for the final decision."

MUJTAHID:
"Shura is not a vote. It is a structured consultation to uncover blind spots and weigh evidence. The Prophet (peace be upon him) consulted his companions even when revelation was available. Why? To sharpen his understanding of context.

Methodology:
- *Istisharat* (seeking counsel) from three groups: experts (finance, engineering, Islamic law), stakeholders (CEO, user advocates), and those affected (users).
- *Tarjih* (weighing): When opinions conflict, give more weight to the *daleel* that is *qati* (definitive) over *zanni* (probable). For example, the user's emotional response ('I feel exploited') is *zanni* but must be taken seriously. The Quranic prohibition is *qati*. So if a pricing model *feels* exploitative to a significant user segment, it likely violates the *maqsad* even if technically permissible.
- *Tadwin* (recording): Document every opinion with the reasoning. If you reject an opinion, write why. This protects you from later *ghurur* (arrogance) and allows for *muraja'a* (revision).

After shura, you should have:
- A clear *shart* (condition) for the pricing model (e.g., 'Must not exceed 20% churn in first 90 days').
- A *munkathir* (nullifier) statement: 'If user fairness perception drops below 70%, roll back.'
- A prioritized list of *zanni* assumptions to test in the next sprint."

10%, rollback to cost-plus and iterate on value messaging.

Rollback trigger: If churn exceeds 7% in the first week, pause the test and revert to previous pricing.

MUHASABA (RETROSPECTIVE)

What would it take for us to cut our price by 50% and still sustain the business? If we can't answer that question honestly, we haven't truly discovered the value we deliver — we're just guessing. A just price isn't the highest the market will bear; it's the price that reflects genuine worth while preserving the relationship. Did our pricing increase trust or erode it? Did we protect the user's wealth or exploit their need? The answer will surface in the churn rate and the unsaid resentment in every cancelled subscription.

Lesson 8. Growth Metrics — Barakah vs Vanity

Vanity metrics count seeds that never sprout. Barakah* growth is increase that carries blessing — sustainable, user-centric, tied to long-term retention and family continuity (*Hifz al-Nasl*). This lesson builds the Barakah Filter and the Barakah ratio you apply to every growth experiment before it ships.*

Maqsad: Hifz al-Nasl (Preservation of Lineage/Continuity) → Sustainable Growth

---

1. THE SCENARIO

You're the Product Lead at a fast-growing fintech startup. Your CEO bursts into your office: "Why are we spending two sprints on a retention feature when we could be acquiring new users through paid ads? We need to hit 500K signups by Q4." You open your notebook. The **Mujtahid** in you asks: *What is the hukm of this growth? What is the maqsad? Is this Barakah or vanity?* The **Product Lead** in you pulls out the AARRR funnel and an Opportunity Solution Tree. You know the CEO is chasing a vanity metric. But you need evidence—and a ruling—to redirect the ship. This is Principle #8.

---

2. DISCOVERY (ISTIQSA')

PRODUCT_LEAD:
Let's run Continuous Discovery. First, map the Opportunity Solution Tree. The desired outcome is *sustainable growth*—users who stay, invite their families, and build long-term financial health. The opportunity is: *Users who reach their first savings goal have 80% retention at 6 months*. The problem we're solving is not "more signups" but "more users who complete the activation loop."

We use Jobs-to-be-Done: What job is the user hiring our product to do? For our fintech app, the core job is *"Help me save consistently for my family's future without stress."* That's **Hifz al-Nasl**—preserving continuity of lineage by protecting wealth and family stability. Growth that undermines that job (e.g., incentivizing debt or gambling-like features) is not Barakah.

So our discovery question: *What does growth with Barakah look like?* We interview 10 long-term users. They say: "I only recommend this app to my siblings because it actually helped me stop late fees." "I don't care about the referral bonus—I just want my brother to get out of debt too." These users are our North Star. Their behavior shows that Barakah growth is **organic, trust-based, and aligned with family well-being**.

MUJTAHID:
Istiqsa' (exhaustive inquiry) begins with defining the problem space through the lens of **Maqasid al-Shariah**. The maqsad here is **Hifz al-Nasl**—preserving lineage and continuity. For a fintech product, this means the growth strategy must not lead to harm that breaks families (debt cycles, gambling, exploitation of the vulnerable).

Vanity metrics—total signups, raw MAU, paid acquisition numbers—are *gharar* (uncertainty) because they don't reflect genuine benefit. They are like counting seeds that never sprout. Barakah growth is *increase that carries blessings*—it multiplies benefit without corrupting the intention.

We must map the **problem space** using the five Maqasid as filters:
- **Hifz al-Din** – Does this metric encourage ethical behavior?
- **Hifz al-Nafs** – Does it protect users from financial stress?
- **Hifz al-Aql** – Does it promote sound decision-making?
- **Hifz al-Mal** – Does it preserve wealth (not waste it on acquisition)?
- **Hifz al-Nasl** – Does it sustain family continuity?

Acquisition-only metrics fail Hifz al-Mal (waste) and Hifz al-Nasl (no continuity). Retention and referral metrics that come from genuine value—those have Barakah. Our istiqsa' reveals that the real opportunity is not "more users" but "more users who stay and bring their family." That is growth with Barakah.

---

3. EVIDENCE (ISTIDLAL)

PRODUCT_LEAD:
Let's gather daleel—evidence. Quantitative: We pull cohort analysis. Users acquired through paid ads have a 30% 90-day retention. Users acquired through organic referrals have 70%. The median lifetime value (LTV) of a referred user is 3x higher. The paid users churn faster and cost more to acquire. That's quantitative daleel that paid acquisition is *less Barakah* for this product.

Qualitative: User interviews reveal a pattern. "I downloaded because of a discount code, but I never used it after the first month." "My sister told me this app helped her save for hajj—so I trust it." The qualitative daleel confirms: Barakah growth is rooted in trust and real outcomes, not incentives.

We also look at *negative evidence*: churn spikes after promotional campaigns. Users who joined via paid ads are 50% more likely to delete the app within 7 days. This is *sad al-dhara'i*—blocking the means to harm. The harm here is wasted resources (israf) and misleading metrics that cause product teams to optimize for the wrong thing.

MUJTAHID:
Now we apply the **daleel hierarchy** from Usul al-Fiqh.
- **Qati al-Thubut** (certain in transmission) – We don't have revelation, but we have reliable analytics data from our own backend. That is strong evidence, albeit zanni (speculative) in interpretation.
- **Qati al-Dalala** (certain in meaning) – The meaning is clear: referred users stay longer. That's a *qiyas* (analogy): if organic referral leads to higher retention, then investing in referral features is analogous to preserving wealth (Hifz al-Mal) and lineage (Hifz al-Nasl).
- **Istishab** (presumption of continuity) – We presume that the current pattern (paid users churn) will continue unless proven otherwise. So we should not scale paid acquisition without stronger evidence.

- **Maslaha** (public benefit) – Sustainable growth benefits the community: users save money, families are less stressed, the company survives long-term. Vanity growth harms the company (burnout, wasted budget) and users (poor experience).

What is "churn" in spiritual terms? Churn is *inqita'*—a severing of the relationship. The user leaves the blessing. A metric that measures "users who leave" is good, but more important is "users who stay and refer." That is *iltizam*—commitment. The Barakah filter asks: *Does this metric measure genuine commitment or fleeting attention?*

Evidence conclusion: The daleel stack points away from paid acquisition as a primary growth lever. The strongest evidence (quant + qual + qiyas) supports investing in referral programs, activation improvements, and retention loops. Those metrics survive the Barakah filter.

---

4. SHURA

PRODUCT_LEAD:
Time for Shura—structured consultation. I schedule a 90-minute cross-functional sync. Attendees: CEO, VP Marketing, Head of Engineering, Customer Support Lead, and two user researchers. I frame the decision: *We need to choose between doubling down on paid acquisition or building a referral + retention engine.*

First, I present the evidence from Discovery and Istidlal: the cohort data, user interview clips, and the Barakah filter. Then I open the floor.

- **CEO** argues: "We need scale to raise Series B. Paid ads are proven. Referrals take time."
- **VP Marketing** pushes back: "Our paid CAC has risen 40% this quarter. The quality is dropping."
- **Customer Support Lead** shares: "Most complaints come from paid users who don't understand the product."
- **User researcher** shows a clip: "I only used it because of the $5 bonus—now I forgot I even have the app."

I weight the inputs based on proximity to evidence and expertise. The CEO's opinion is important but based on a fear of missing growth targets. The Support Lead and researcher have direct daleel from users—their voices carry more weight. I record all opinions, including dissent.

MUJTAHID:
Shura in Usul is not a democratic vote. It is a structured process to surface the strongest daleel and reach a ruling that serves the maqsad. The methodology:
1. **Identify the stakeholders** – Everyone present has a right to speak, but we weight their *wilaya* (authority) based on knowledge and proximity to the truth. The user researcher and support lead are the *shuhud* (witnesses) to reality.
2. **Record each opinion with its daleel** – CEO's daleel: "We need speed to survive." That is a maqsad of Hifz al-Mal (company survival), but it conflicts with Hifz al-Nasl (sustainable growth). We note the tension.
3. **Seek consensus (ijma')** – We don't have full consensus. The CEO dissents. We document

that dissent as *mukhalafa* (opposition) but proceed based on the preponderance of evidence.
4. **Issue the ruling** – The Shura concludes## THE PRINCIPLE (HUKM)

HUKM: We will only ship growth initiatives that pass the **Barakah Filter** — meaning the metric must be sustainable, user-centric, and directly tied to long-term retention (Hifz al-Nasl), not just short-term acquisition or vanity spikes.

DALEEL:
- Quantitative: Cohort analysis shows that users acquired via "growth hacks" (discounts, viral loops, referral spam) have a 60% lower 90-day retention than organic users.
- Qualitative: User interviews reveal that aggressive referral prompts erode trust — "I felt like a product, not a person."
- Qiyas (analogy): Just as riba (usury) creates apparent wealth but destroys real economic stability, vanity metrics create apparent growth but destroy product lineage (Nasl). The Barakah filter is analogous to *sad al-dhara'i* — blocking the means to harm, even if the means look beneficial in the short term.

MAQSAD: Hifz al-Nasl (Preservation of Lineage / Continuity) — ensuring the product grows in a way that sustains its user community and value over generations, not just this quarter.

SHURUT:
- Each growth experiment must define a **Barakah ratio**: (Retained users after 30 days) / (Total users acquired). Target: ≥ 40% for any experiment to continue.
- All growth tactics must be **opt-in, transparent, and avoid manipulation** (no dark patterns, no pre-checked boxes, no shame-based prompts).
- Monthly Barakah audit: Review AARRR funnel through the lens of *maslaha* (public benefit). If any stage has a negative *maslaha* score (e.g., activation causes user anxiety), the initiative is paused.
- Rollback trigger: If support tickets from growth features exceed 2% of new users within 7 days, automatically halt the experiment.

MUNKATHIRAT:
- **Nullifier one:** Any growth tactic that violates user data privacy or consent (even if legal) — this nullifies the principle immediately. *Hifz al-'aql* (preservation of intellect) requires informed choice.
- **Nullifier two:** If the growth metric improves but the core product experience (NPS, task success rate) declines more than 5% relative to baseline — the principle is broken.
- **Nullifier three:** If the team cannot articulate how the growth initiative serves a real user job (JTBD) beyond the company's revenue goal — the principle is suspended until a clear *maqsad* is defined.

THE PROTOCOL (3 STEPS – THIS SPRINT)

STEP 1: Audit Your Current Metrics

Monday–Tuesday.
Pull your last 3 months of AARRR data. For each stage, write two numbers: the raw metric

(vanilla) and the **Barakah-adjusted metric** (retention at 30 days ÷ acquisition cost or effort). Highlight any stage where the Barakah ratio is below 40%. Flag those as "vanity growth." Discuss with your team: "Which of these are we secretly proud of but actually harming our Nasl?"

STEP 2: Build and Calibrate Your Barakah Filter

Wednesday–Thursday.

Create a simple 3-question filter for every growth experiment before it ships:

1. Does this serve a real user job (JTBD) or just our revenue goal?
2. Can a user opt out without penalty?
3. Will this increase *thiqah* (trust) or just traffic?

If any answer is "no," the experiment is blocked. Write the filter as a checklist in your PRD template.

STEP 3: Apply the Filter to One Upcoming Growth Experiment

Friday.

Take your next planned growth initiative (e.g., a referral program, a discount campaign). Run it through the Barakah filter. If it passes, define the specific *shurut* (conditions) and *munkathirat* (rollback triggers) for that experiment. If it fails, redesign it — or kill it. Ship only if it passes. Measure the Barakah ratio after 30 days.

MUHASABA (RETROSPECTIVE)

Where are we currently optimizing for a vanity metric that we know — deep down — is misleading our team and eroding our product's long-term trust?

Sit with that question for 10 minutes. No defensiveness. Write down the metric name. Then ask: *What would it cost us to drop this metric entirely and replace it with a Barakah-adjusted one?* The answer is usually far less than the cost of chasing ghosts.

Lesson 9. Autonomous Squads with Embedded Shura

Scaling judgment means distributing it. This lesson organizes teams into autonomous squads that own a full opportunity space and decide through structured shura* rather than hierarchy — the prophetic Medina model of small, consulting units (*nuqaba'*) applied to a modern product org. (The opening of this principle is partial in the source; the surviving Shura, Principle, Protocol, and retrospective are preserved faithfully.)*

CEO's solution. But CEO's outcome (reduce fraud complaints) is valid. The squad's *what* should be: "Help users self-audit transactions to reduce fraud calls." That serves the CEO's outcome and the user's job.

MUJTAHID:

Shura in Islam is not a vote. It is *istishara* — seeking counsel from those with *'ilm* and *hilm* (wisdom). The *mujtahid* (decision-maker) listens to all, then issues a *hukm*. Dissent must be recorded but does not block action.

Record the **dissent** from the Engineering Lead: “We will lose velocity building fraud alert first.” That becomes a *shart* (condition) for the decision: if velocity drops below X, revisit.

Now document the *shura* process in a log:
- Participants: CEO, Eng Lead, PM, User Researcher.
- Inputs: Outcome (reduce complaints), User JTBD (self-audit), Technical constraints (2 weeks vs 3 months).
- Weighted evidence: User research (2x weight), Engineering feasibility (1.5x), CEO outcome (1x).
- Decision: Squad *maqsad* = “Reduce fraud complaints by enabling self-audit via transaction history.”
- Dissent recorded: Eng Lead wants fraud alert; rationale noted. Condition: If after 4 weeks fraud complaints don’t drop, re-open shura.

This is *shura at scale* — not a committee of equals, but a structured consultation where the *wali al-amr* (Product Lead) holds final authority, bounded by the *maqsad* and *shurut*.

HUKM: We will organize our product teams into autonomous squads with embedded Shura circles, where each squad owns a full opportunity space and makes binding decisions through structured consultation, not hierarchy.

DALEEL: Three evidence streams converge. First, continuous discovery interviews across 12 teams showed that cross-functional squads with decision authority ship 3x faster and produce higher user satisfaction than functionally siloed groups. Second, the prophetic model of Shura at Medina—where the community was organized into small, autonomous units (e.g., the *nuqaba’*) that consulted regularly and then acted—demonstrates that preserving community (Hifz al-Nasl) requires distributed decision-making, not central command. Third, our own A/B test of squad vs. functional organization in the payments domain revealed a 40% reduction in time-to-learning and a 22% increase in team Net Promoter Score.

MAQSAD: Hifz al-Nasl (Preservation of Community) is served directly. Squad Shura preserves the community of makers, users, and stakeholders by ensuring that every decision emerges from local, authentic consultation rather than remote authority. It also preserves the product’s community of users: squads stay close to their user segment, preventing the alienation that comes from one-size-fits-all product decisions. Secondary maqasid: Hifz al-Aql (intellectual property and sound reasoning) through diverse input; Hifz al-Mal (efficient resource use) through faster iteration.

SHURUT: - Each squad must include at least one product manager, one designer, and two to three engineers (cross-functional minimum). - Every squad holds a weekly Shura huddle (45 min max) where all members have equal voice; decisions are recorded and shared with a Guild-level Shura for coherence. - Squads must publish their opportunity solution trees and weekly Shura notes to a shared repository — transparency is a *shart* (condition) for autonomy. - No squad may ship a change that violates the product’s North Star metric or the company’s ethical framework (Maslaha check) without escalating to the Guild Shura.

MUNKATHIRAT: - If a squad's opportunity solution tree goes unvisited by any user interview for two consecutive sprints, the Shura circle is invalidated and must be reconvened with fresh evidence. - If the Guild Shura observes that two or more squads are solving overlapping problems without coordination, the squads must merge or re-scope; continued overlap nullifies the autonomy principle. - Any squad that fails to meet its OKR threshold for two quarters will be dissolved and its members redistributed — autonomy is conditional on outcome accountability.

THE PROTOCOL

STEP 1: Map your current org to the Org Quadrant (this sprint).
Draw four boxes: Functional (teams by skill), Cross-Functional (teams by feature), Squad (teams by opportunity space), Guild/Chapter (cross-squad communities). Plot every team you have. Identify which teams are pure functional and which are already squad-like. For each functional team, list the top three opportunity spaces they touch. This map is your baseline. Deliverable: a single A3 page with the quadrant and annotations.

STEP 2: Select one opportunity space and form a pilot squad (next sprint).
Pick the opportunity space with the highest user pain and the clearest JTBD (Jobs-to-be-Done). Pull one PM, one designer, two engineers from existing functional teams. Give them a single OKR: "Improve [metric] by X% in 8 weeks." Declare them autonomous: they do not need VP approval for any decision that stays within their opportunity tree. They must, however, hold a Shura huddle every Wednesday at 10 AM sharp. No exceptions.

STEP 3: Run the pilot for 8 weeks, then run a Muhasaba retrospective.
The pilot squad ships at least one experiment per week. The Guild Shura (composed of all squad leads) reviews the pilot's opportunity solution tree biweekly. At week 8, answer: "Did the squad preserve community? Did it ship faster? Did users feel heard?" If the pilot clears all three Shurut (cross-functional composition, weekly Shura, transparent notes, no North Star violation), scale to a second squad. If any Munkathirat fire (e.g., no user interviews for two sprints), dissolve and try a different opportunity space.

MUHASABA (RETROSPECTIVE)

What did we actually fear when we gave a squad full autonomy, and how much of that fear was rooted in evidence versus ego?

This question cuts to the heart of Hifz al-Nasl. Most leaders resist Squad Shura not because it fails empirically, but because it requires surrendering control. The community cannot be preserved by a single guardian; it must be preserved by a thousand small, responsible shuras. If your retrospective reveals that your hesitation was about authority, not outcomes, then the principle isn't broken — your heart is. Fix that first.

Lesson 10. Lasting Impact — Sadaqah Jariyah & Impact ROI

The capstone. A product that dies when you stop paying attention is sadaqah 'adiyah* (one-time charity); a product designed to keep benefiting people without you is *sadaqah jariyah* (on-going charity). This lesson measures Impact ROI and generational compounding, validating the mechanism by analogy (*qiyas*) to *waqf*. The real test: if you died tomorrow, would this product continue to benefit people?*

$50/month have 6x lifetime value. Users who refer three friends have 2x retention. But what's the *compound interest* on impact? Measure net promoter score (NPS) for "trust that this product serves my long-term good." That's your leading indicator.

Qualitative: Run a diary study. Ask 20 users to log every time they feel their money is "doing good" beyond themselves. Code the responses. Themes emerge: *"I want my savings to mean something after I'm gone," "I'd pay a small fee if it went to a cause I believe in," "I don't trust the bank to do good—I want control."*

MUJTAHID:
Daleel hierarchy: We need evidence that this feature is *maslaha* (public benefit) and not *munkar* (harm).

1. **Qati al-Thubut, Zanni al-Dalala:** The hadith on sadaqah jariyah is authentic (qati al-thubut) but its application to product features is interpretive (zanni al-dalala). So we need *qiyas* (analogy).
2. **Qiyas:** Compare a digital product that automates charitable giving to a physical endowment (waqf). The `'illah` (effective cause) is *continuity of benefit*. If a waqf building generates rent for a school, a product feature that auto-distributes micro-donations from savings generates ongoing benefit.
3. **Istihsan (juristic preference):** Allow innovation in delivery mechanism as long as the maqsad (preservation of mal and din) is served.
4. **Maslaha Mursalah (unrestricted benefit):** This feature does not contradict any text; it aligns with the maqsad of preserving wealth (mal) by making it fructify in the akhirah.

Evidence Prompt Applied:
What is the "interest rate" on product impact? Calculate **Impact ROI**:

```
Impact ROI = (Total benefit accrued to users over 10 years) / (Cost to build + maintain)
```

Example: A $50k feature that enables 10,000 users to each give $100/year to charity for 10 years = $10M in sadaqah. Impact ROI = 200x. That's compound.

Measure generational impact: Track *second-degree referrals*—users whose parents used your feature. If a user sets up a legacy account, and their child later uses the same feature, that's generational compound.

Synthesis:
Evidence supports building a "Legacy Savings" module. Quantitative data shows users with long-

term trust stay longer. Qualitative data shows desire for afterlife impact. Islamic daleel validates the mechanism as sadaqah jariyah via qiyas on waqf. The `'illah` is continuity. The maqsad is preservation of din and mal.

---

4. SHURA (CONSULTATION)

PRODUCT_LEAD:
Call a cross-functional sync. Invite Engineering, Legal, Compliance, and Customer Support.

- **Engineering:** "Can we build a rule engine that lets users set recurring charitable allocations from savings? Yes, but we need API integration with Islamic charities."
- **Legal/Compliance:** "Are there regulatory implications for 'estate planning' features? We need to avoid giving financial advice. Frame it as a 'giving tool' not a will."
- **Customer Support:** "Users will call confused. We need clear onboarding and a fatwa verification badge."

Interview three religious scholars (or Shari'ah advisors) separately. Ask: *"Is this a valid form of sadaqah jariyah? What conditions must we meet?"* One scholar says: "The user must intend sadaqah at the time of setting up. The charity must be reputable. The mechanism must not involve riba or gharar."

MUJTAHID:
Shura in Usul al-Fiqh is not a vote; it's a consultation that uncovers blind spots. The Mujtahid must weigh input by *adalah* (integrity) and *khibra* (expertise).

- Weight the scholar's input highest (expertise in Shari'ah).
- Weight engineering input second (feasibility).
- Weight legal input third (regulatory risk).

Record dissent. One compliance officer says: *"This blurs the line between product and philanthropy. We might be seen as a non-profit."* The Mujtahid notes this as a *shubha* (doubt) to address via *sad al-dhara'i* (blocking the means to harm). Solution: Clear disclaimers that the feature is a tool, not a charitable institution.

Shura Prompt Applied:
How do you consult? Schedule a 90-minute session. Open with the maqsad: *"We want to build a product that becomes sadaqah jariyah. What do we need to make it valid, trustworthy, and scalable?"*

- Read aloud the hadith on sadaqah jariyah.
- Show the qiyas to waqf.
- Ask each stakeholder: *"What could go wrong? What are we missing?"*

Close by summarizing three **shurut** (conditions) that emerged from shura:
1. User must make intention (niyyah) at setup.

2. Charity recipient must be verified.
3. No riba in the underlying savings product.

30%, the teach loop is viable.

MUHASABA (RETROSPECTIVE)

Where is your product currently acting more like a *sadaqah 'adiyah* (one-time charity) than a *sadaqah jariyah* (ongoing charity)?
Be honest: list every feature that dies when you stop paying attention. Then ask yourself: *If I died tomorrow, would this product continue to benefit people without me?* If the answer is no, you have not built a legacy — you have built a job. The challenge is not to add more features, but to design the exit. A *Sadaqah Jariyah* product is one that makes its creators unnecessary. That is the ultimate measure of compound impact.

Worksheet — The Decision Memo Template

Use this template to turn any live product decision into a principled ruling. It mirrors the structure every lesson in this course builds toward. Fill each field honestly; a blank field is a signal that you are not yet ready to decide.

1. The Mas'ala (the question). State the decision as a question, not a foregone conclusion. *Example: "Should we ship instant payouts next sprint?"*

2. The Job-to-be-Done. What job is the user actually hiring the product to do? Not the feature — the struggle and the progress sought.

3. Decision type (Decision Quadrant). Is this Reversible or Irreversible? High stakes or Low stakes? Your type sets your evidence bar: reversible decisions tolerate *zanni* (probable) evidence; irreversible ones demand *qati'* (definitive).

4. The Daleel (evidence). List the evidence across the quadrant and mark each item's strength:

Source	Quadrant (Quant / Qual / Behavioral / Attitudinal)	Strength (qati' / zanni)	What it indicates

5. The Maqsad (purpose). Which higher objective does this decision serve — *Hifz al-Din* (purpose), *al-Nafs* (safety/wellbeing), *al-Aql* (clarity/judgment), *al-Mal* (wealth/resources), *al-Nasl* (continuity/community)? Name the primary and any secondary.

6. The Shura (consultation). Who did you consult, what did each say, and how did you weight it (by expertise and proximity, not rank)? Record dissent explicitly — dissent becomes a condition or a nullifier.

7. The Hukm (the ruling). State the decision in one sentence.

8. The Shurut (conditions). What must be true for this ruling to hold? *Example: "Launch at 100% but behind a kill-switch, with a $200/day payout cap."*

9. The Munkathirat (nullifiers / rollback triggers). What observations invalidate the ruling and force a rollback? *Example: "If support tickets exceed 2% of new users within 7 days, halt."*

10. The Muhasaba (retrospective). Set the date you will revisit this memo, and write the one question you will ask yourself then — the question you would least like to answer.

End of course. May your judgment be sound and your intentions pure.

— Falah Digital