How does Bitcoin evolve and what is its roadmap?

Luke Dashjr

Ethical considerations

Who needs to agree?

- Does it deprive others of their property/bitcoins? (censorship)
- Does it create/increase burden(s) on others? (block size increases)
- How does it affect people who don't participate?

If everyone needs to agree, we need strict consensus!

Whether hardfork or softfork doesn't matter for this.

Ethical considerations

Does it make sense to exclude people?

- Irrational objections
- Saboteurs
- People undermining the network security? (eg, no full node of their own)

Technical considerations

What is technically needed for a change to successfully be deployed?

Note, these considerations are in addition to ethical considerations.

Technical considerations: Layer 2

What is technically needed for a change to successfully be deployed?

With layer 2, users can just choose what to use on a case-by-case, person-by-person basis. No consensus is needed at all. If two people want to use it, they can, without permission or adoption from anyone else.

- Original L2: p2p flood network & pay-to-IP
- Before long, people moved to Bitcoin addresses (still w/ flood net)
- To avoid stuck transactions, RBF was adopted as a change to flood net
- Lightning replaces flood network & addresses with more direct p2p & payment channels

Technical considerations: Softforks

What is technically needed for a change to successfully be deployed?

With a layer 1 protocol change, consensus of some form is needed. Softforks are accepted by default: if you do nothing, you remain on the upgraded network. (If the community doesn't want it, we can still opt-out!)

For Bitcoin to be secure, however, most people must use their own full node! Softforks degrade former full nodes to light nodes. (Remember, Bitcoin is not a system where we just trust miners.)

Softforks need user nodes updated, not just miners nodes.

Technical considerations: Hardforks

What is technically needed for a change to successfully be deployed?

By default, all nodes reject hardforks. It will fail unless everyone explicitly opts-in by upgrading. A hardfork is basically an airdropped altcoin proposed as a replacement for the old system.

With careful planning, most hardforks can be made slightly "softer" so that old nodes neither accept nor reject them. With this, users must make an explicit decision one way or the other.

Technical considerations: Extension blocks

What is technically needed for a change to successfully be deployed?

A hybrid between softforks and hardforks is the extension block. This kind of change degrades not only the security of old nodes, but also the functionality. They require a lot of technical complexity and carry a lot of technical debt.

They do, however, behave similar to softforks: unless you act to reject it, you will end up accepting it implicitly.

How to measure consensus

Positive, strict consensus in a large decentralised community is an unsolved problem. (Simple hardforks at least may be impractical.)

When there isn't consensus, it is usually obvious. Unpopular proposals tend to have widespread objections, and even if a smaller portion of the community objects, that minority tends to be loud about their objection.

If there's no apparent objection to a widely publicised proposal, we can probably at least assume that nobody will actively choose to opt-out.

What kind of change?

- Minimise disruption.
- Maximise probability of success.
- Avoid technical debt and/or complexity.
- Avoid unnecessary trust.
- Prefer layer 2, then softfork, soft-hardfork, hardfork, extension block.

What kind of change?

Examples:

- Lock times are based on blockchain properties, so can't go in layer 2.
- Confidential transactions fundamentally changes the consensus logic for checking that transactions aren't giving out more bitcoins than they spend, so it cannot be done as a softfork.
- Extension blocks can slightly reduce the friction to deploying mere block size increases (by making the default opt-in), but at a large complexity and technical debt cost. It is better therefore to use a [soft-]hardfork.
- Fundamental changes to the UTXO model such as MimbleWimble cannot reasonably be done without an extension block.

Process of making a change

1. Float the idea with the community
2. Get developer agreement on a specific solution
   (Including a safe deployment method!)
3. Write a draft BIP (Bitcoin Improvement Proposal) - implement & review
4. Measure community support - check that it isn't likely to fail
5. Merge implementation to major node software (including old versions)
6. Deploy - make sure release notes are clear to users

Possible future changes (roadmap)

- Segwit v1 - revised Script language; simpler signatures; sign-time Script
- Lightning - real p2p transactions using less on-chain space and instant
- Signature aggregation - reduces transaction sizes and verification time
- Confidential transactions (maybe not enough privacy?)
- Decentralised sidechains - perhaps revisit if mining gets less centralised
- Blockstream's Simplicity - safer smart contracts w/ turing-like flexibility
- Reducing the block weight/size limit - making Bitcoin sustainable

How does Bitcoin evolve and what is its roadmap?

Luke Dashjr