Section outline

2. CRAMES: Compressed RAM for embedded systems
   - Introduction, motivation, and past work
   - CRAMES design
   - Compression algorithm design
   - Experimental evaluation
   - Commercialization

Problem background

RAM quantity limits application functionality

RAM price dropping but usage growing faster

Secure Internet access, email, music, and games

How much RAM?

- Functionality
- Cost
- Power consumption
- Size

Ideal hardware–software design process

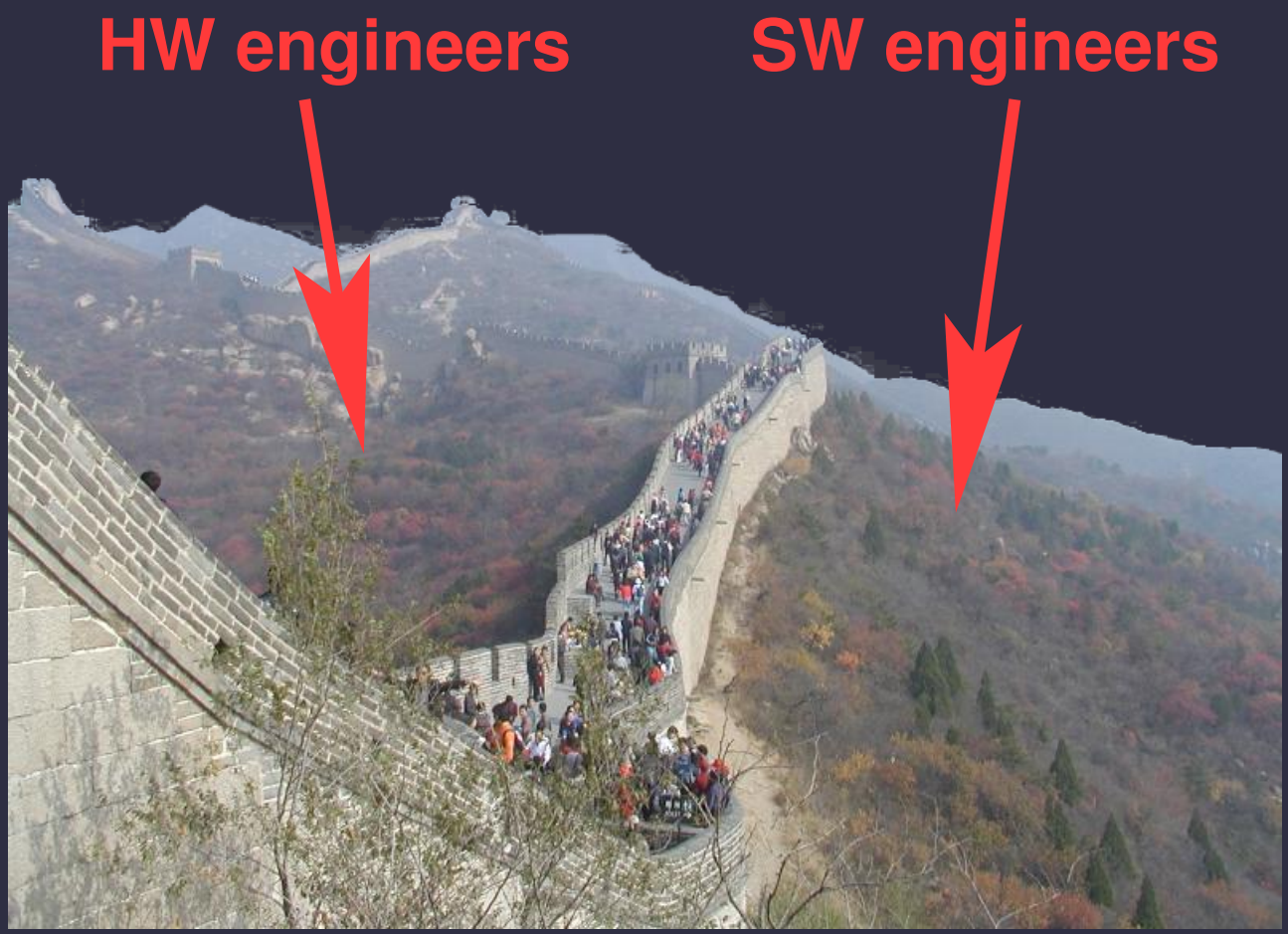
Ideal case

Hardware and software engineers collaborate on system-level design from start to finish

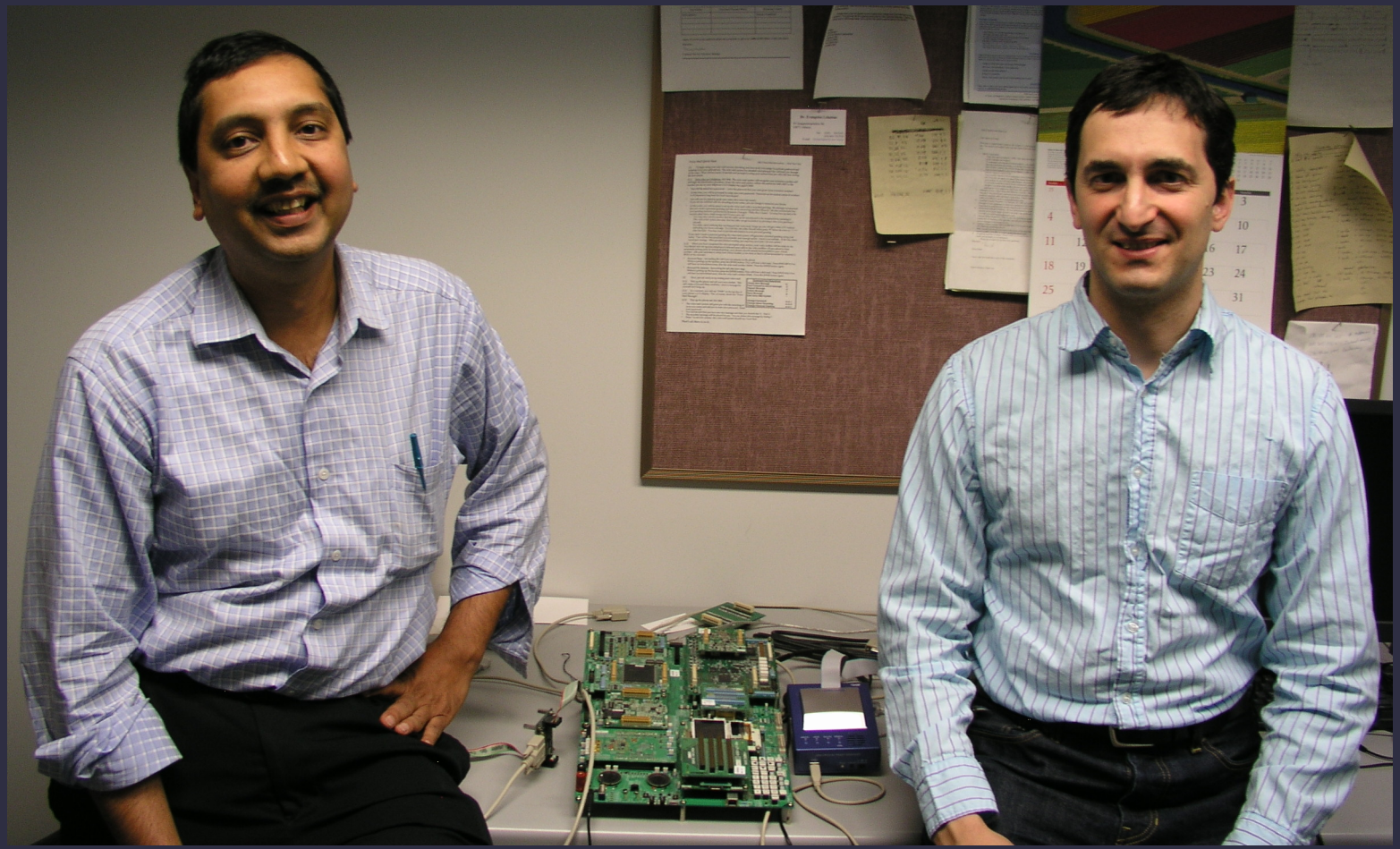
We teach the advantages of this in our classes

It doesn't always happen

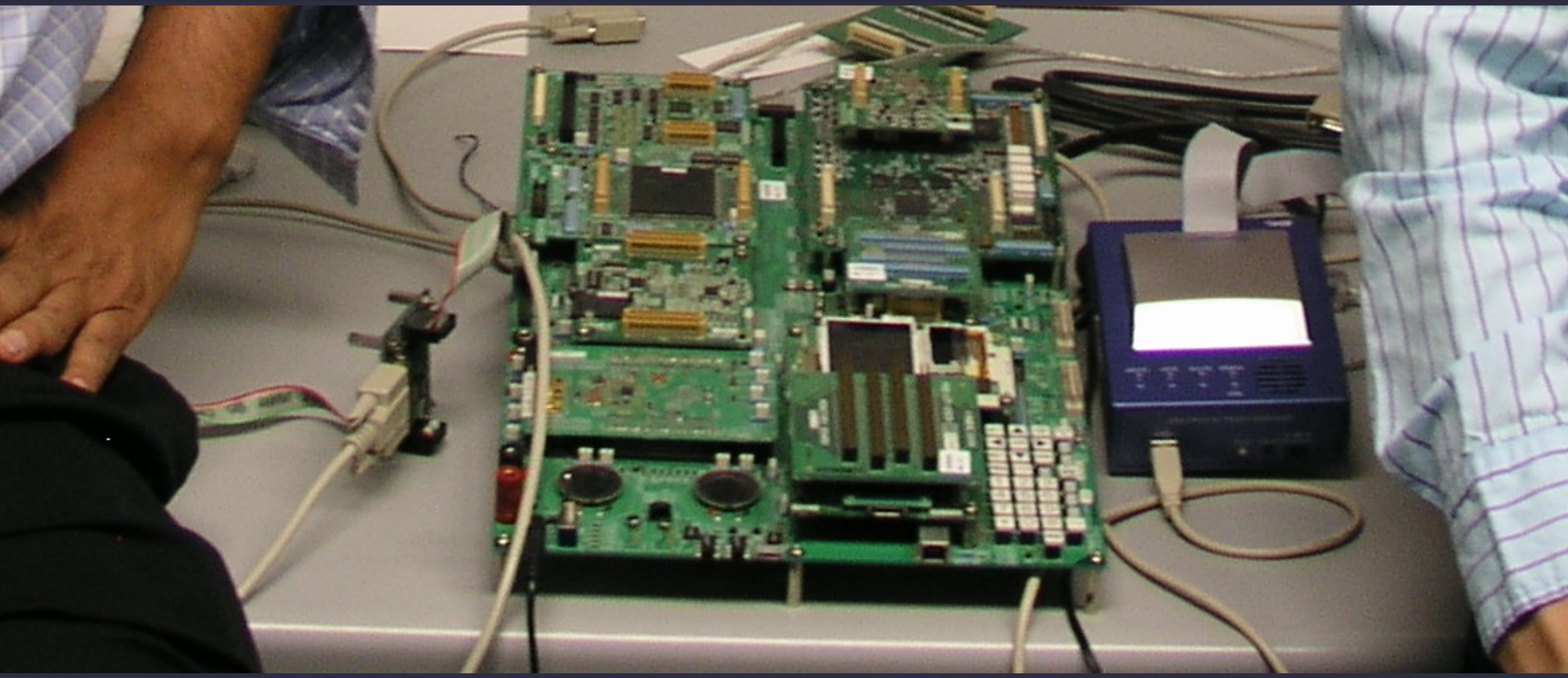
Real hardware–software design process



NEC collaborators and cellphone prototype



NEC collaborators and cellphone prototype



Options

Option 1: Add more memory

Implications: Hardware redesign, miss shipping target, get fired

Option 2: Rip out memory-hungry application features

Implications: Lose market to competitors, fail to recoup design and production costs, get fired

Option 3: Make it seem as if memory increased

- Do not change hardware
- Do not change applications
- Do not decrease performance
- Do not increase power

Nobody knew how to do this

Options

Option 1: Add more memory

Implications: Hardware redesign, miss shipping target, get fired

Option 2: Rip out memory-hungry application features

Implications: Lose market to competitors, fail to recoup design and production costs, get fired

Option 3: Make it seem as if memory increased

- Do not change hardware
- Do not change applications
- Do not decrease performance
- Do not increase power

Nobody knew how to do this

Goals

Allow application RAM requirements to overrun initial estimates even after hardware design

Reduce physical RAM, negligible performance and energy cost

Improve functionality or performance with same physical RAM