

# Hva driver vi med på Lysaker? *og hvordan jobber vi*



Olve Maudal, Kunnskapsminister, Cisco Systems in Norway

Bedriftspresentasjon for 1. klasse Elsys, NTNU, 7. Oktober 2015



## About Cisco

Today Cisco are more than 70000 employees with an annual revenue of \$40 billion, which means that Cisco is among the largest global technology companies. Cisco has 20000+ engineers and invests ~13% of revenue on R&D anually.

The headquarter of Cisco is in San Jose, just south of San Fran 

[www.cisco.com](http://www.cisco.com)

Cisco Systems, Innovation Center  
Lysaker, Norway



# Telepresence









Boardroom

Today

Management review

09:30 - 10:30

All hands meeting

17:30 - 18:40

Q2 Finance update

20:00 - 21:00



9:25

1000  
1000





# Some of the stuff we develop at Lysaker



at Lysaker we are ~350 engineers

most of us work with software development



# programming

```
Emacs@Olive-Maudals-MacBook-Pro.local

#ifndef PAL_TYPE2_MESSAGE_HPP_INCLUDED
#define PAL_TYPE2_MESSAGE_HPP_INCLUDED

#include "ntlm_message.hpp"

namespace pal {

    class type2_message : public ntlm_message {
    public:
        explicit type2_message(const std::vector<uint8_t> & buffer);
        virtual std::vector<uint8_t> as_bytes() const;
        uint32_t ssp_flags() const;
        uint64_t challenge() const;
    private:
        const std::vector<uint8_t> buffer_;
    };

}

#endif

```

```

*     targetinfo
*     server (type=0x0100, len, data)
*     domain (type=0x0200, len, data)
*     dnsserver (type=0x0300, len, data)
*     dnsdomain (type=0x0400, len, data)
*     type5 (type=0x0500, len, data) // unknown role
*     <terminator> (type=0, len=0)
*/

pal::type2_message::type2_message(const std::vector<uint8_t> & buffer)
: buffer_(buffer)
{
    enum { min_type2_buffer_size = 32 };
    if (buffer.size() < min_type2_buffer_size)
        throw std::invalid_argument("not a type2 message, message too short");
    const uint8_t prefix[12] = {
        'N', 'T', 'L', 'M', 'S', 'S', 'P', '\0',
        0x02, 0x00, 0x00, 0x00
    };
    if (!std::equal(prefix, prefix + sizeof prefix, buffer.begin()))
        throw std::invalid_argument("not a type2 message, invalid prefix");
}

uint32_t pal::type2_message::ssp_flags() const
{
    enum { ssp_flags_offset = 20 };
    return pal::read_uint32_from_little_endian(&buffer_[ssp_flags_offset]);
}

uint64_t pal::type2_message::challenge() const
{
    enum { challenge_offset = 24 };
    return pal::read_uint64_from_little_endian(&buffer_[challenge_offset]);
}

```

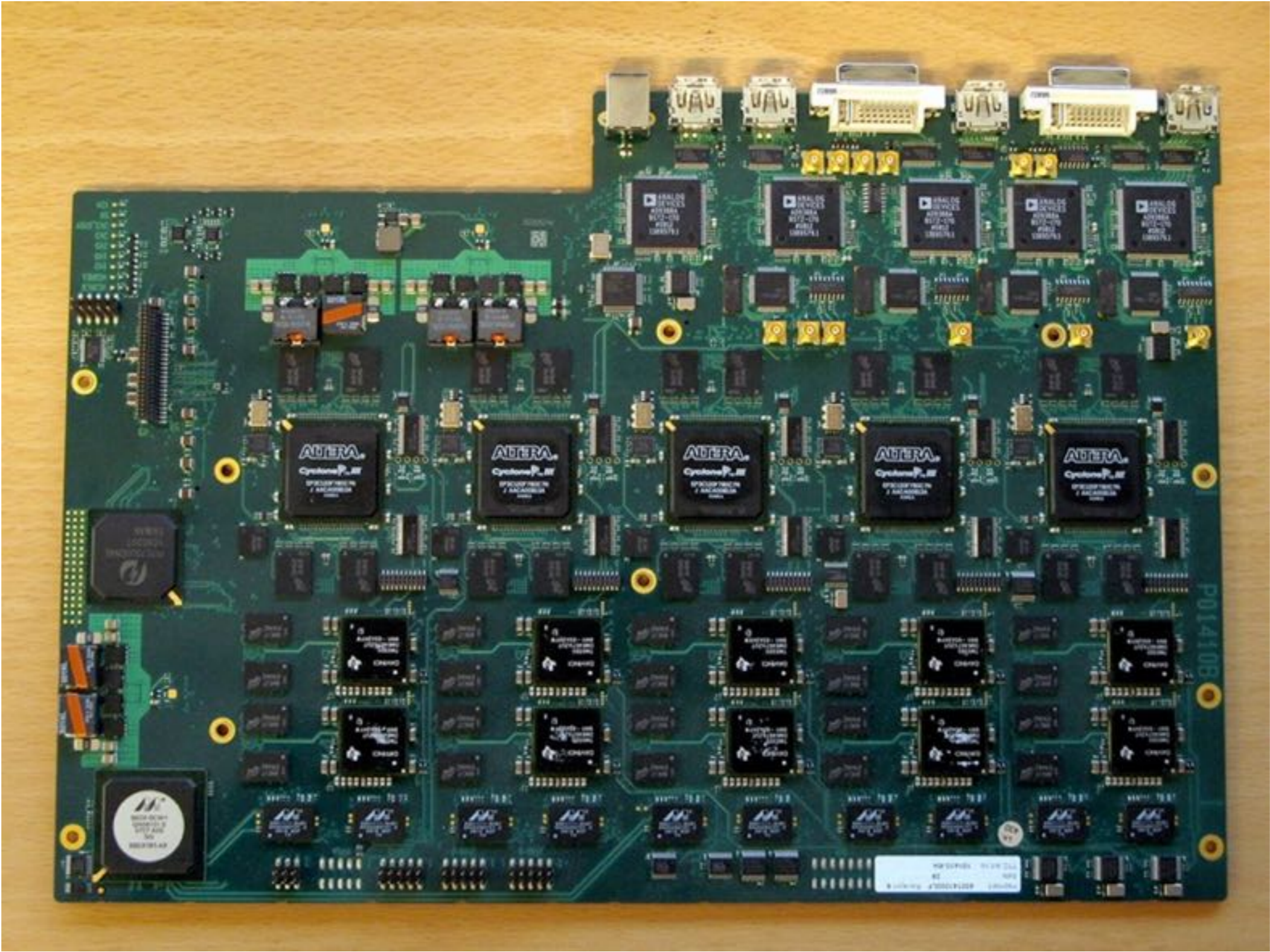
```

--:-- type2_message.hpp All (21,0) Git-master (C++/1 Abbrev)-----:-- type2_message.cpp 37% (45,0) Git-master (C++/1 Abbrev)-----
bash-3.2$ make
g++ -std=c++98 -pedantic -Wall -Wextra -Weffc++ -g -MMD -c -o pal.o pal.cpp
g++ -std=c++98 -pedantic -Wall -Wextra -Weffc++ -g -MMD -c -o type1_message.o type1_message.cpp
g++ -std=c++98 -pedantic -Wall -Wextra -Weffc++ -g -MMD -c -o type2_message.o type2_message.cpp
g++ -std=c++98 -pedantic -Wall -Wextra -Weffc++ -g -MMD -c -o type3_message.o type3_message.cpp
g++ -std=c++98 -pedantic -Wall -Wextra -Weffc++ -g -MMD -c -o tools.o tools.cpp
rm -f libpal.a
ar cru libpal.a pal.o type1_message.o type2_message.o type3_message.o tools.o
ranlib libpal.a
g++ -std=c++98 -pedantic -Wall -Wextra -Weffc++ -g -MMD -c -o tests.o tests.cpp
g++ -o tests -g -lcrypto tests.o libpal.a
./tests
test_byte64_encoding_and_decoding
test_converting_between_hex_string_and_bytes
test_request_challenge_response_sequence
.....
tests OK
g++ -std=c++98 -pedantic -Wall -Wextra -Weffc++ -g -MMD -c -o httpget.o httpget.cpp
g++ -o httpget -g -lcrypto httpget.o libpal.a
bash-3.2$
-l:** 2 Bot (26,10) (Shell:run)-----

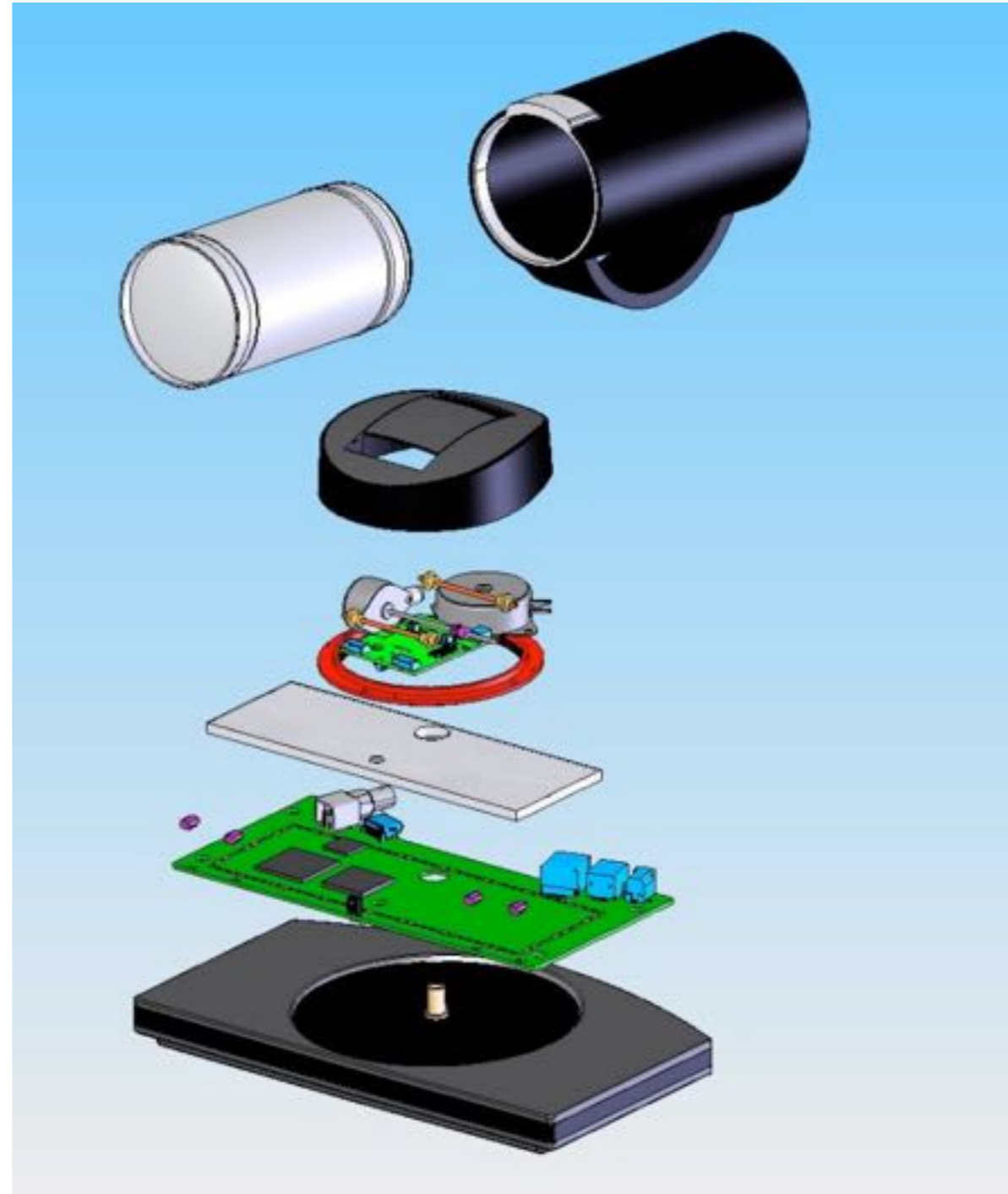
```

but we also do...

# Electronics / Hardware



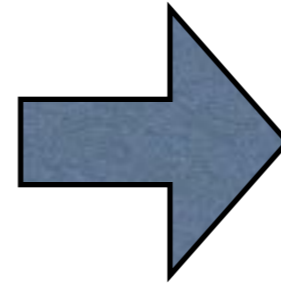
# Mechanics



# Industrial Design and User Experience Design



1992



2015

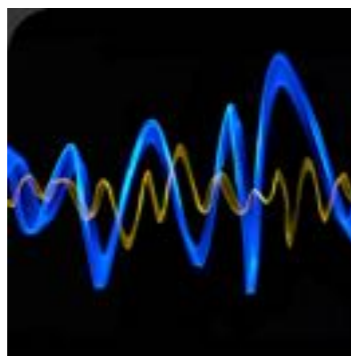


Looking into



the future





Audio / Acustics



Video / codecs



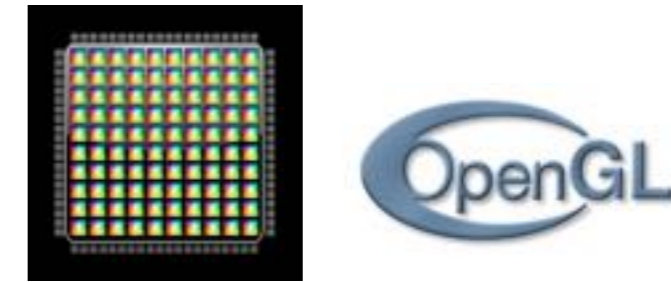
Protocols



Embedded SW



C and C++



Multicore & Graphics



QA & Diagnostics



SW Tools



User Experience Design



Electronics



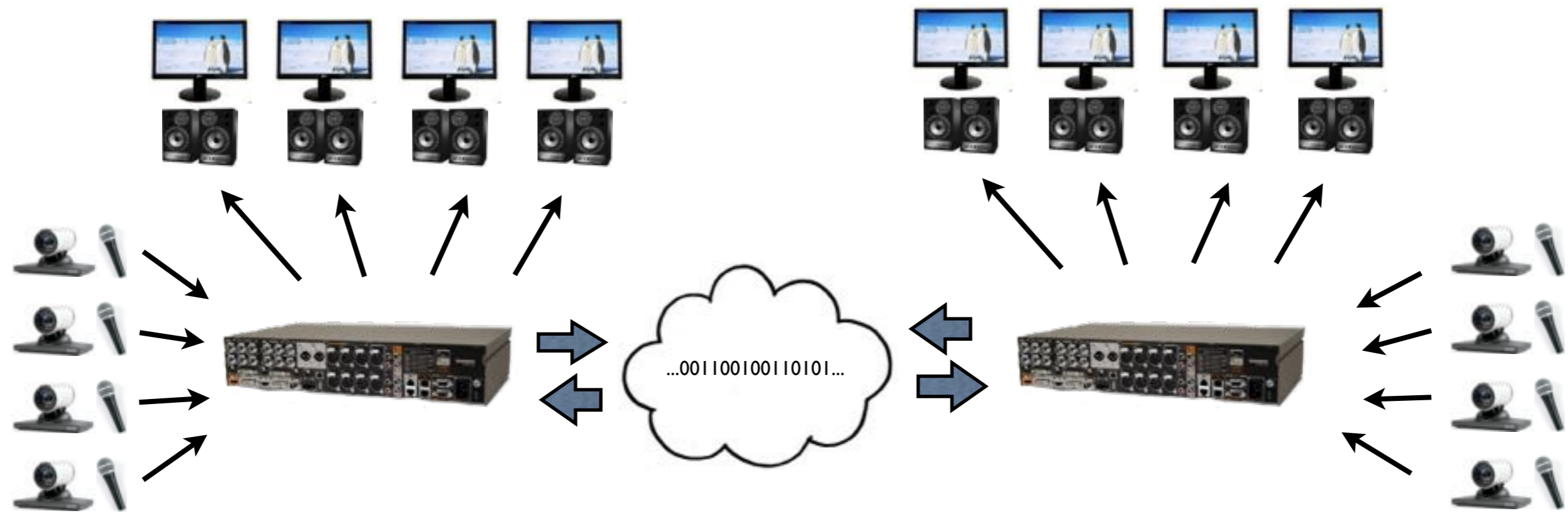
Cybernetics

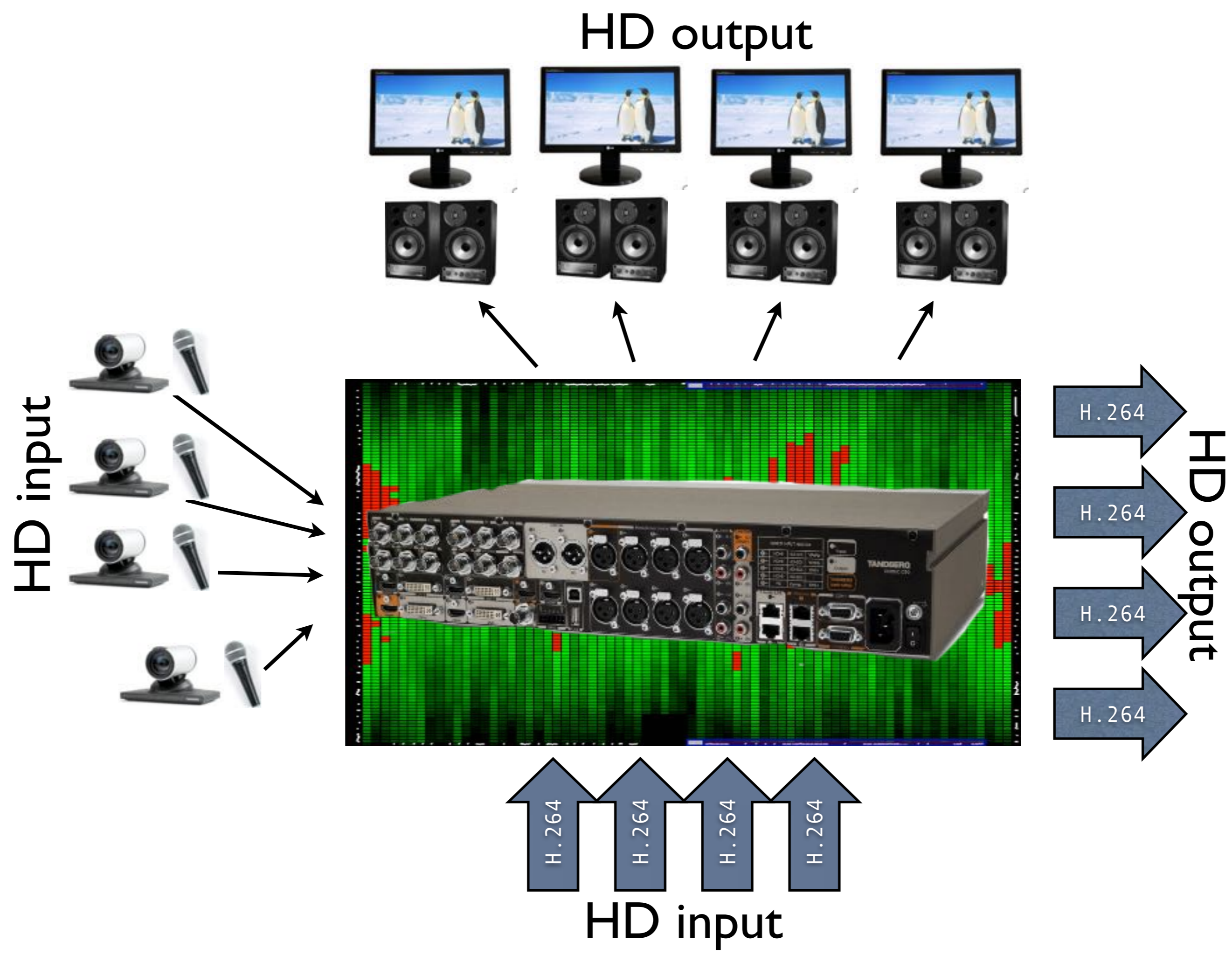


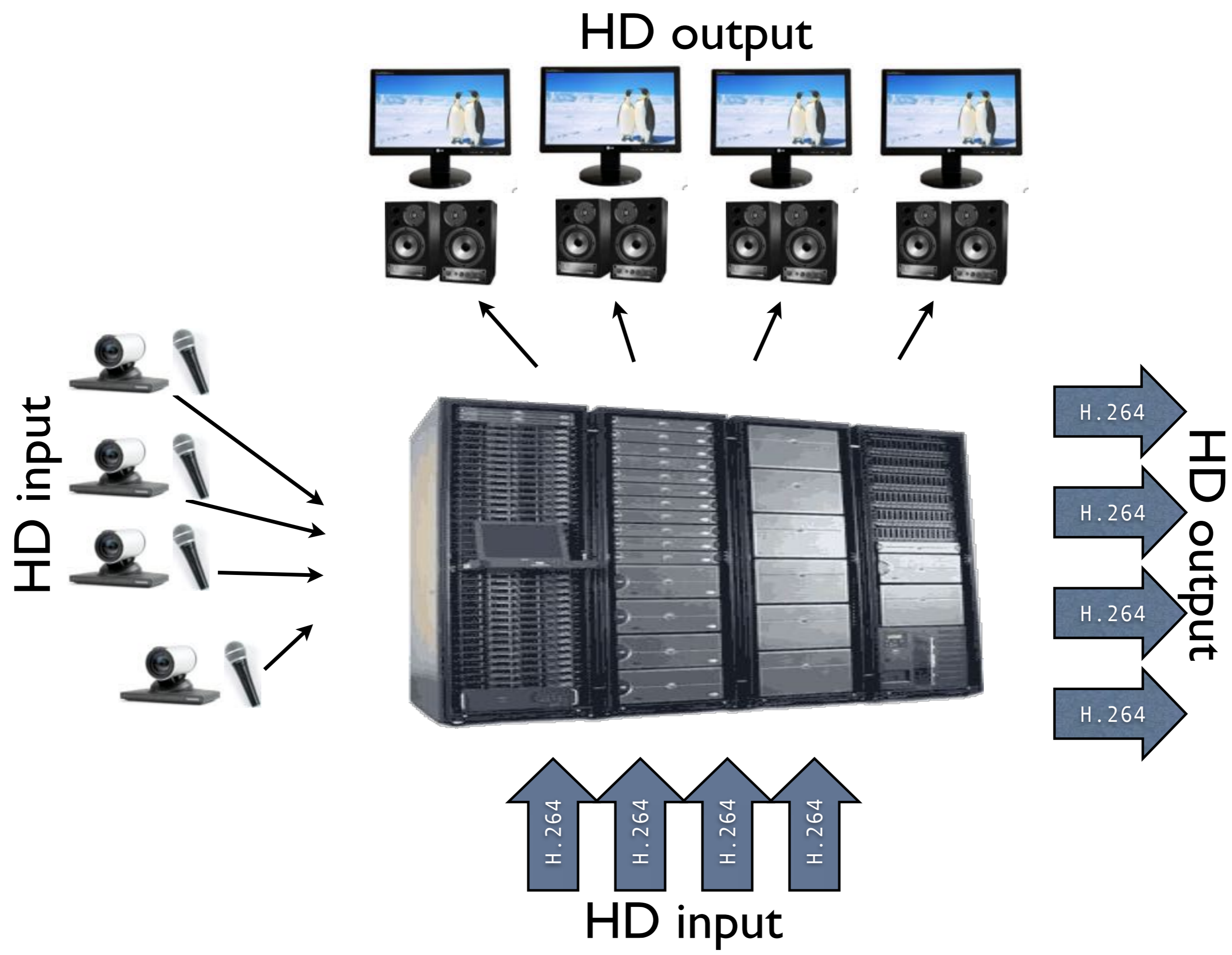
Industrial design and Mechanics

# Example of a Product





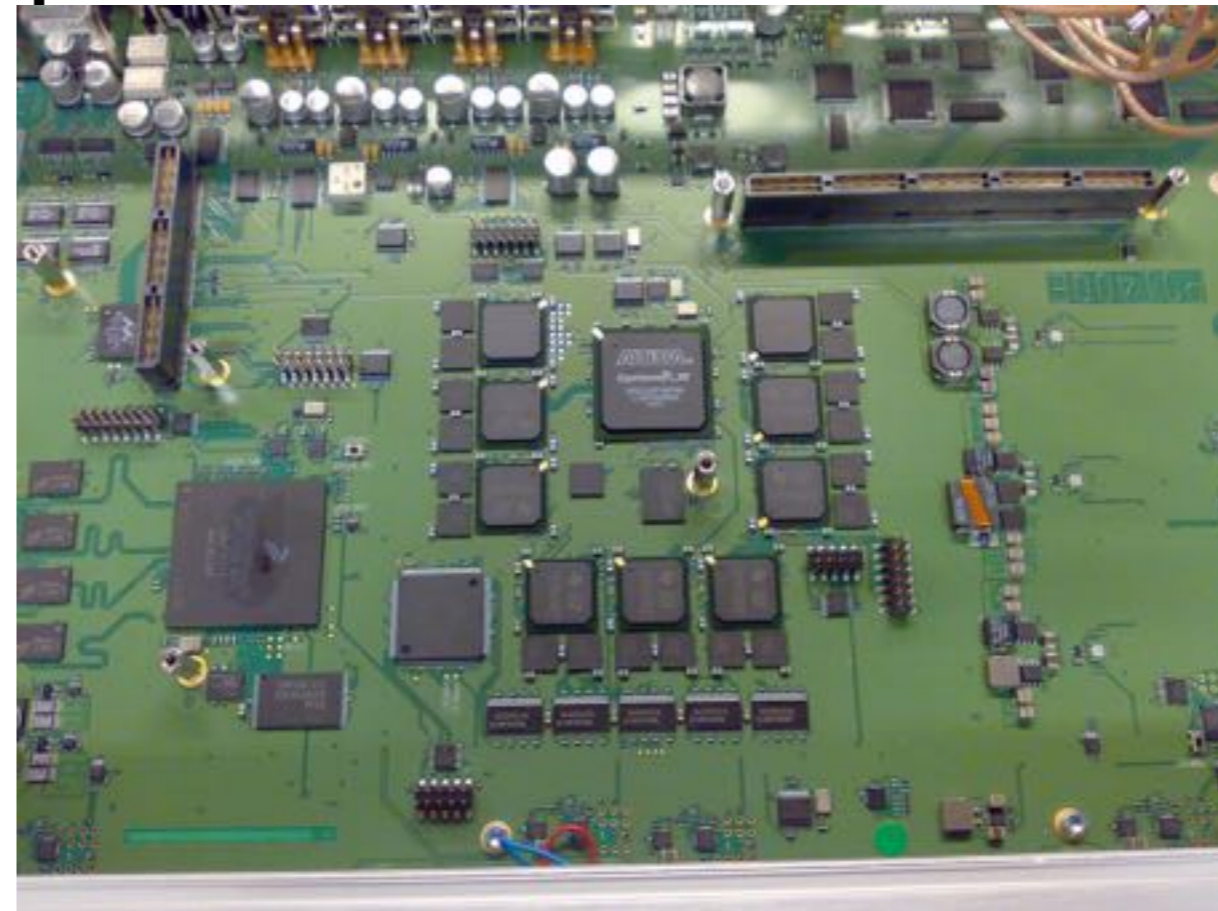






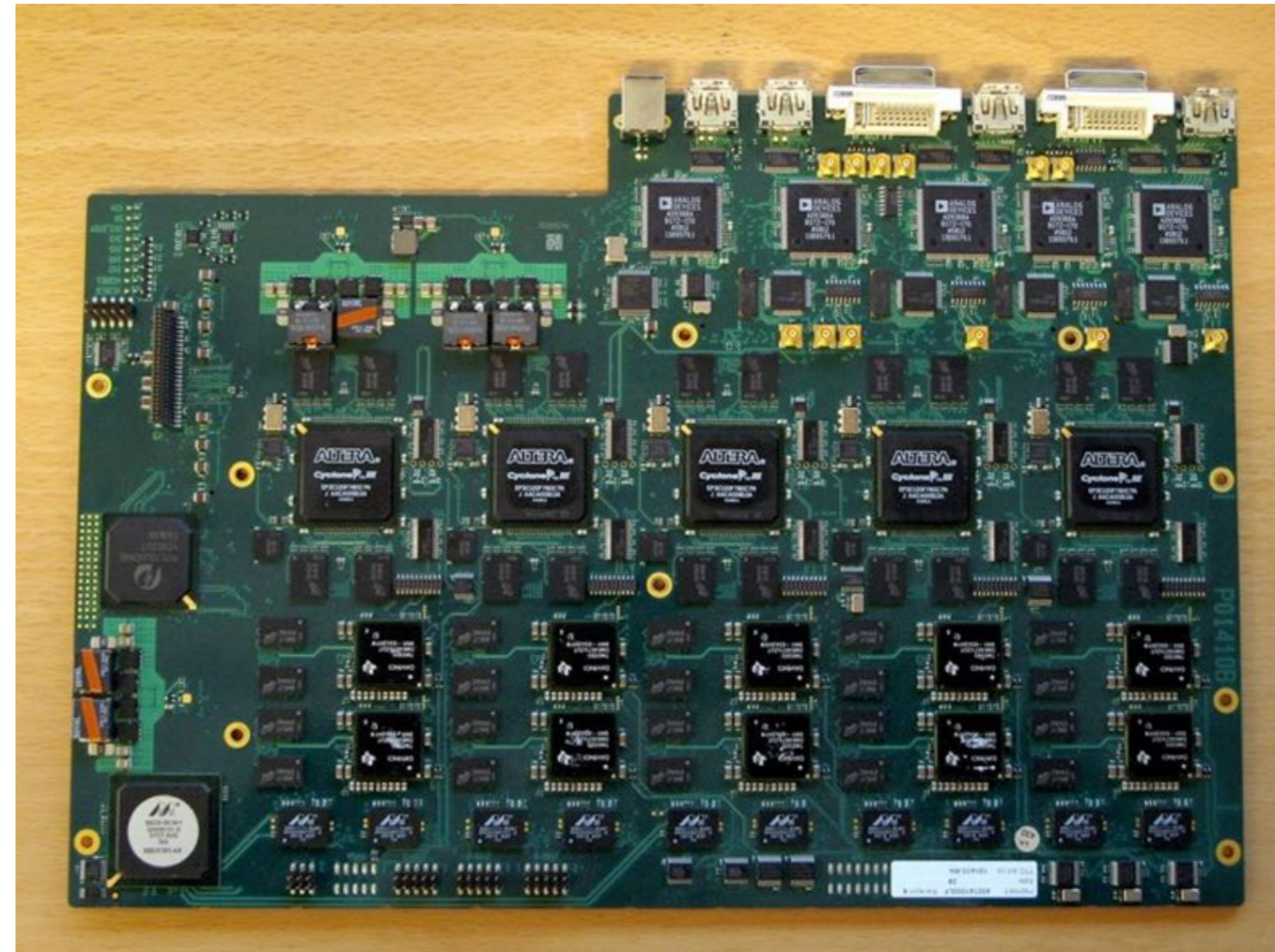
# C90 MAIN BOARD

- 1 Altera Cyclone III 120 for Audio switching (Nios II softcore 50 MHz)
- 9 TI 6727, audio dsp for echo control, compression, decompression, +++
- PowerPC 8347, main processor, application software, networking, user interface
- 3543 components / 15659 pins
- 16 layers
- 3264 nets



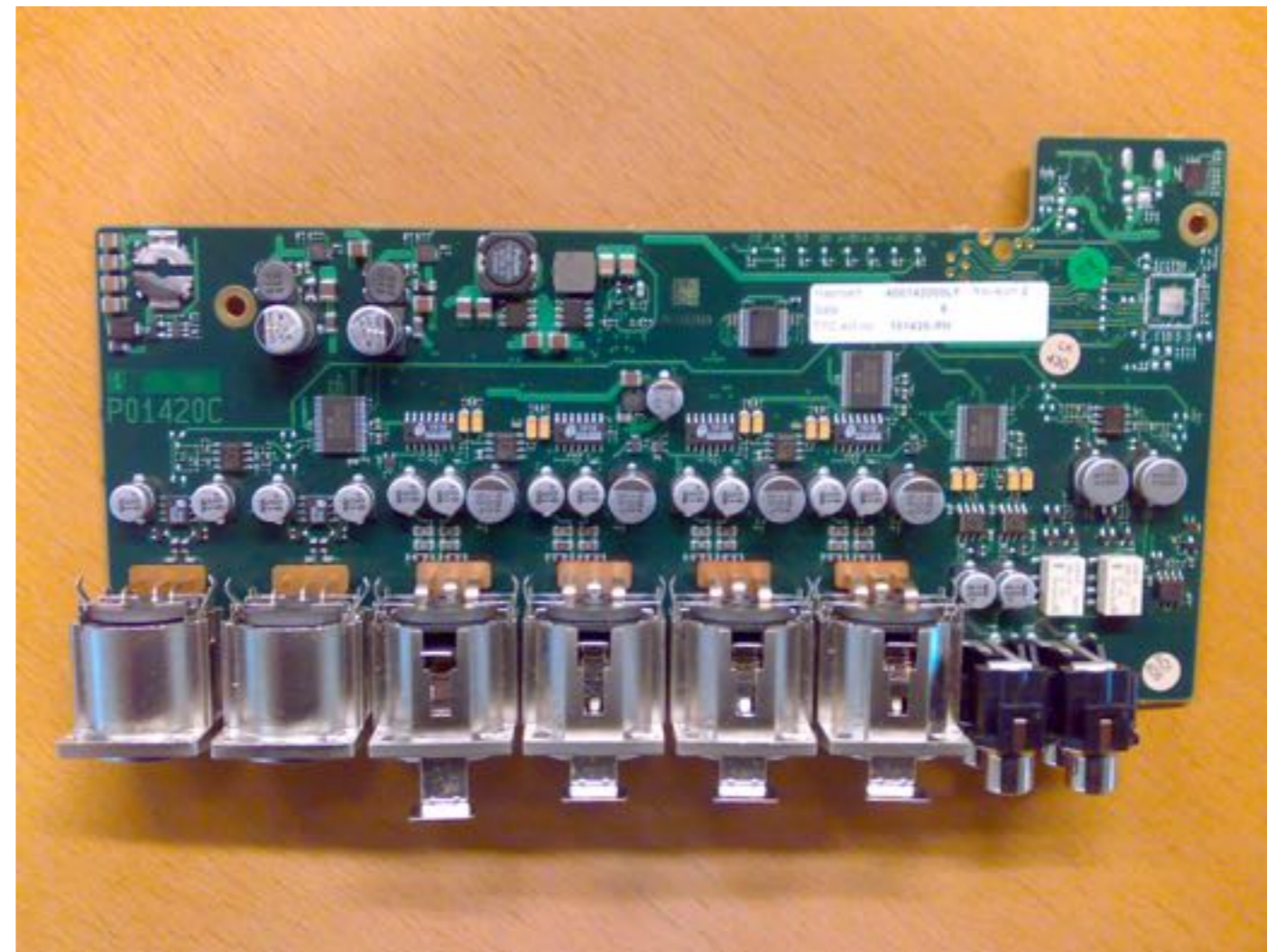
# C90 VIDEO BOARD

- 10 Da Vinci DM6467 for video compression/decompression(1 ARM, 1 dsp, 2 coprocessors),
- 5 Altera Cyclone III 120 for video scaling & composing(Nios II softcore 50 MHz)
- 15 Gbps video backplane
- 3.8 GByte DDR2 RAM
- 128 mbit x5 SDRAM
- 6097 components
- 30520 pins
- 22 layers
- 6490 nets



# C90 AUDIO EXTENSION BOARD

- analog amplification
- high quality AD and DA converters
- pure electronics, no processor/SW
- 717 components
- 6 layers



## **C90 - from a geek point of view**

- 10000+ components
- 44 (6+22+16) layers
- 56 processor cores
- several million lines of code (C and C++)

**Facts about advanced product development**



Few high tech projects are like running down on a paved road where you can see the ...



... goal in the end of the road.

Most projects are more like...



extreme orienteering



in impossible terrain



with a group of people





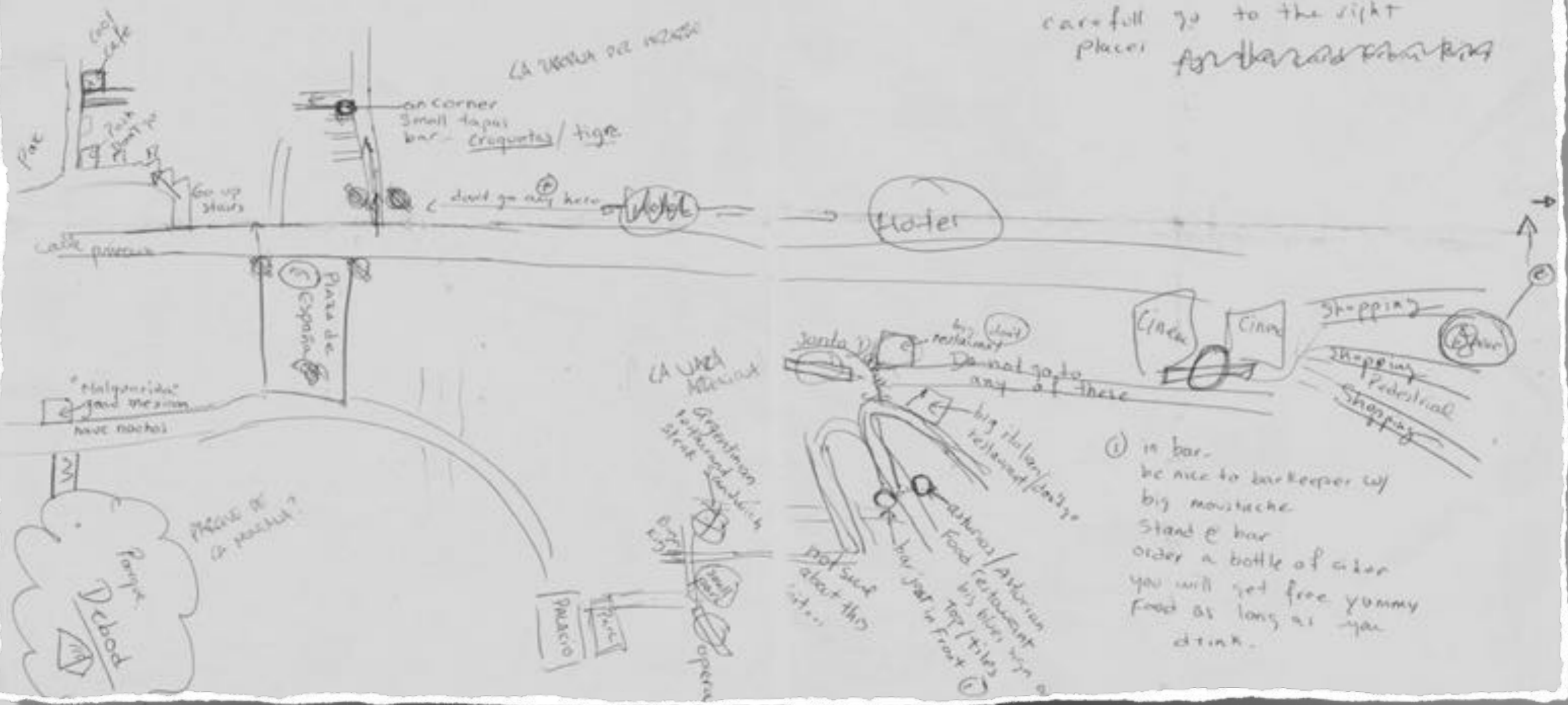
in the dark

Ⓜ Except Museo del Jamon, go there are many in the city if you want a "to go" sandwich, go there for a Jamon y queso with croissant!

with only a sketchy map as guidance

Plaza del sol

Ⓜ cross to the other side and go slightly left. there is a quarter there all restaurant/tapas place - FULL OF ENGLISH & AMERICANS so be careful go to the right place for the red brick building



Ⓜ in bar. be nice to barkeeper w/ big moustache. Stand @ bar order a bottle of cider you will get free yummy food as long as you drink.





*Learn to surf, instead of trying to control the waves...*

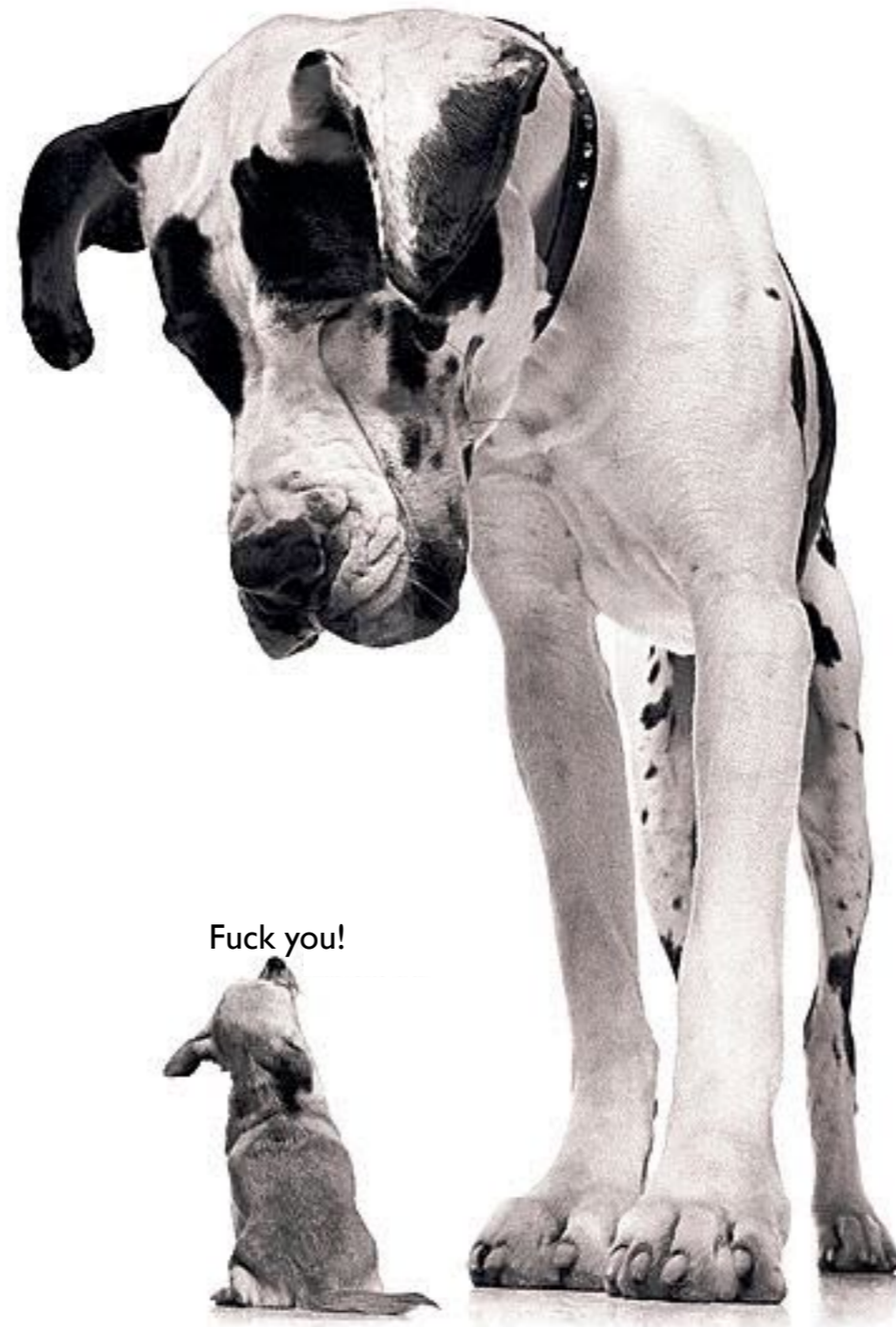
# Some principles of effective product development



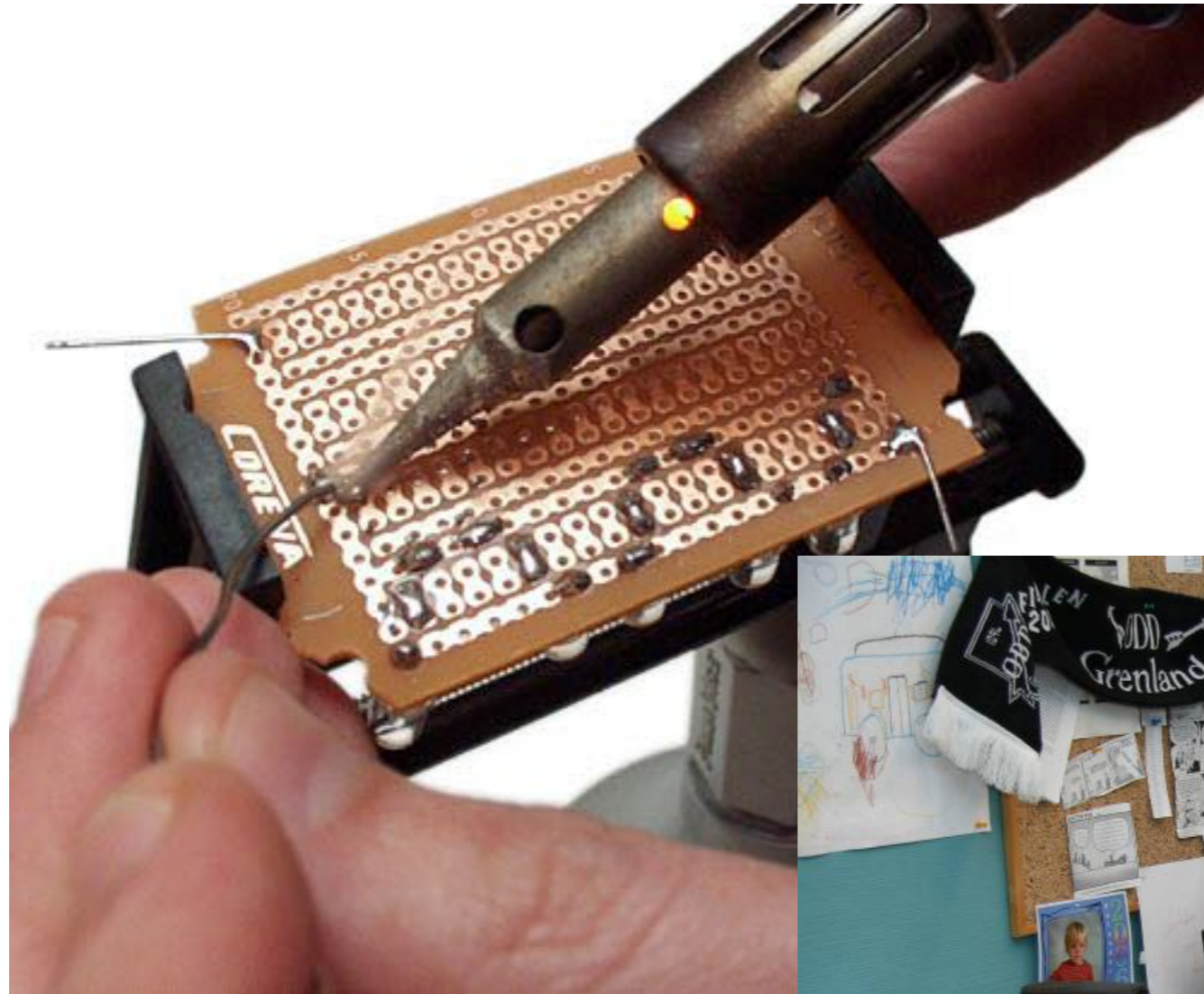
# Embrace chaos



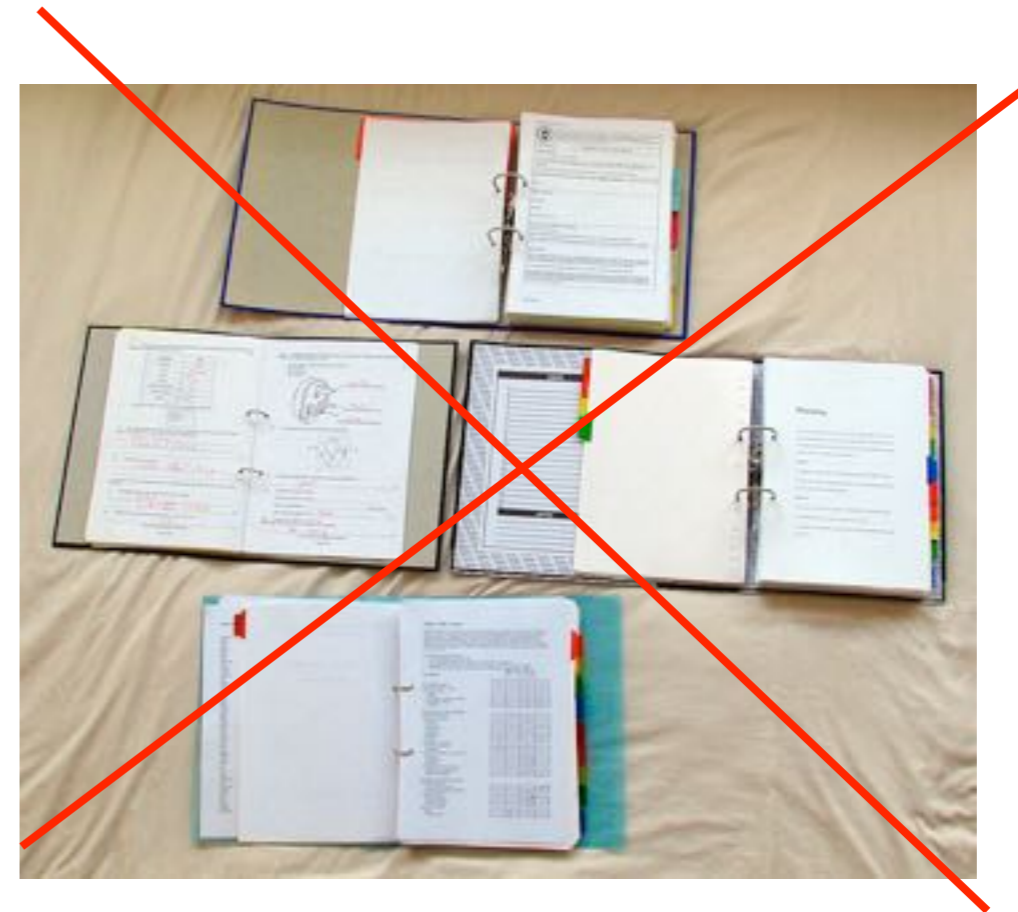
Break the rules



# Respect doers (and create an autonomous organization)



# Focus on communication (over documentation)



# Introduce slack



*If your company's goal is to become fast, responsive, and agile, more efficiency is not the answer—you need more slack. (Tom DeMarco)*



# Beware the observer effect



# Constrain your innovation



## Reward courage (and failures)





Focus on the whole product



system thinking vs reductionism

# Delay decisions (but do continuous planning)

Plans are of little importance, but planning is essential – Winston Churchill

Plans are nothing; planning is everything. – Dwight D. Eisenhower

No battle plan survives contact with the enemy. – Helmuth von Moltke the Elder



Everyone has a plan 'till they get punched in the mouth. – Mike Tyson

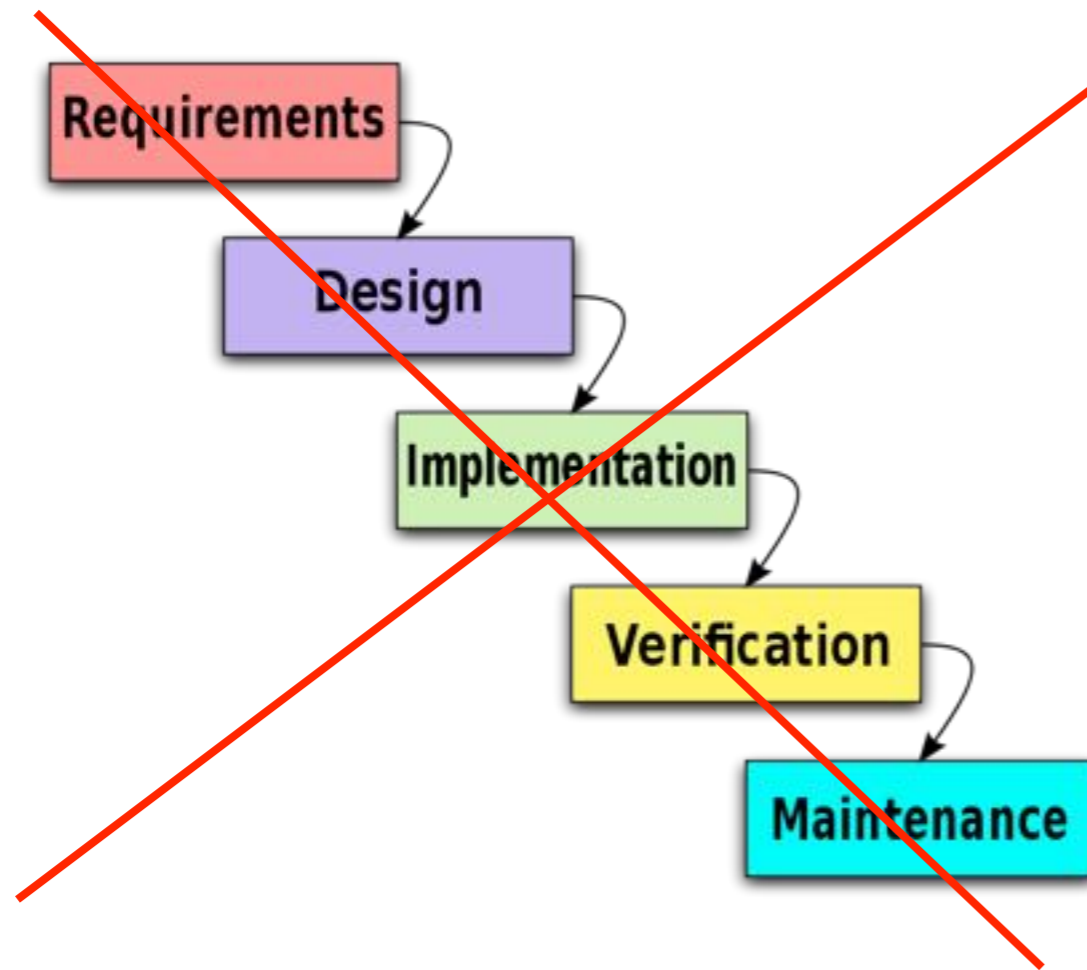
Aim for approximately right rather than accurately wrong



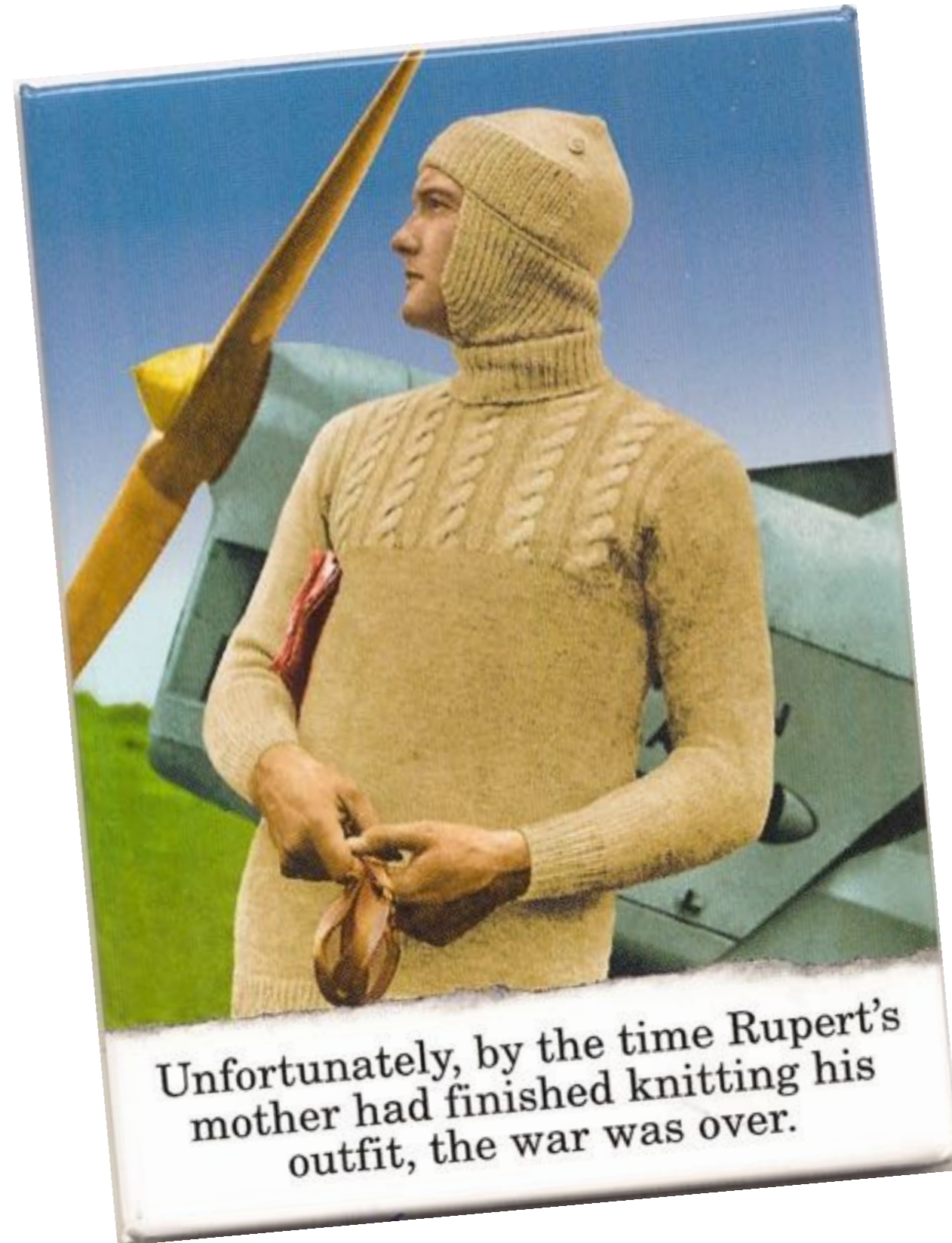
Release early and release often



# Follow principles, not processes



# Timing is everything

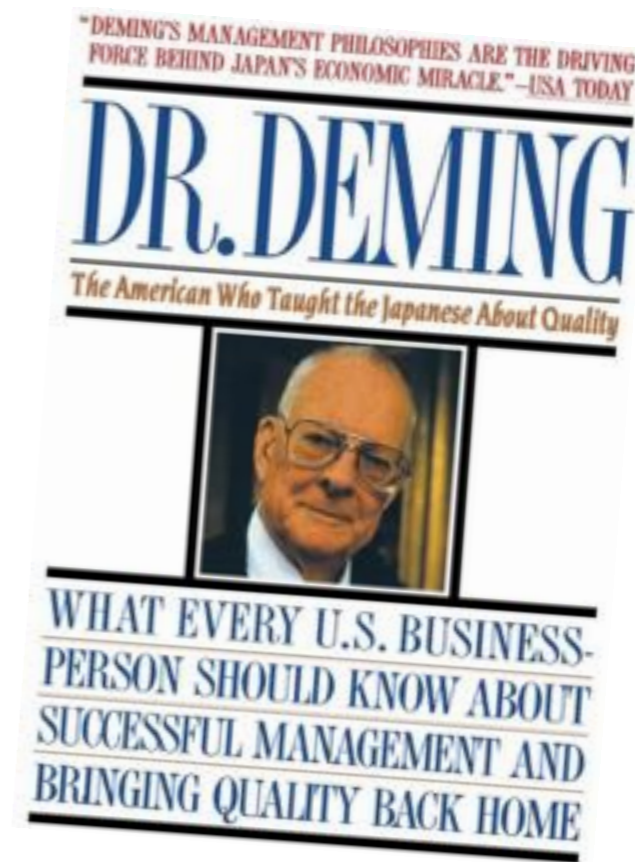
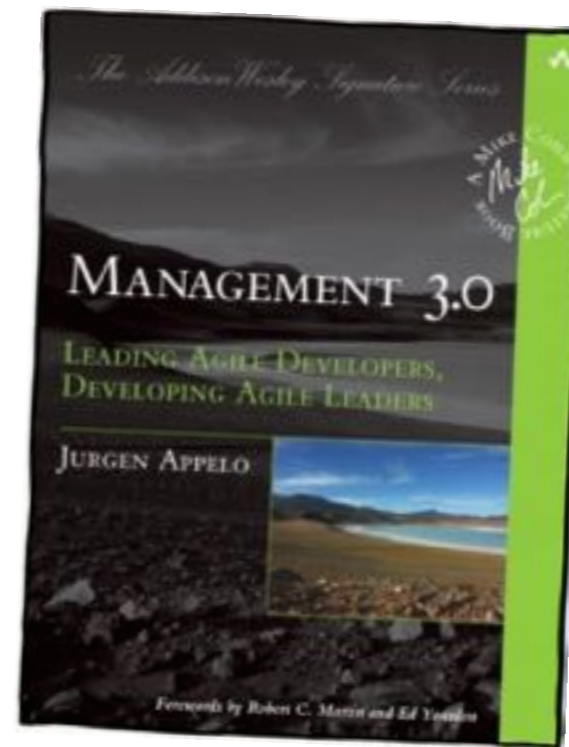
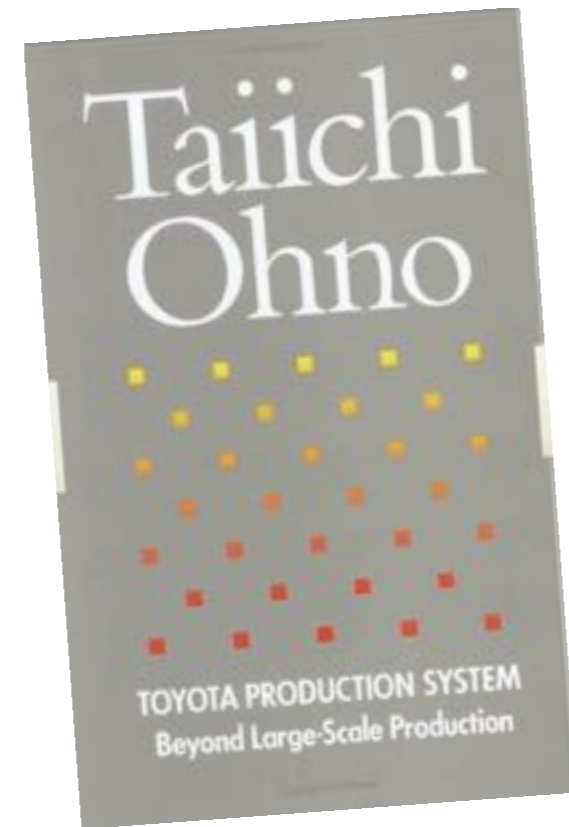
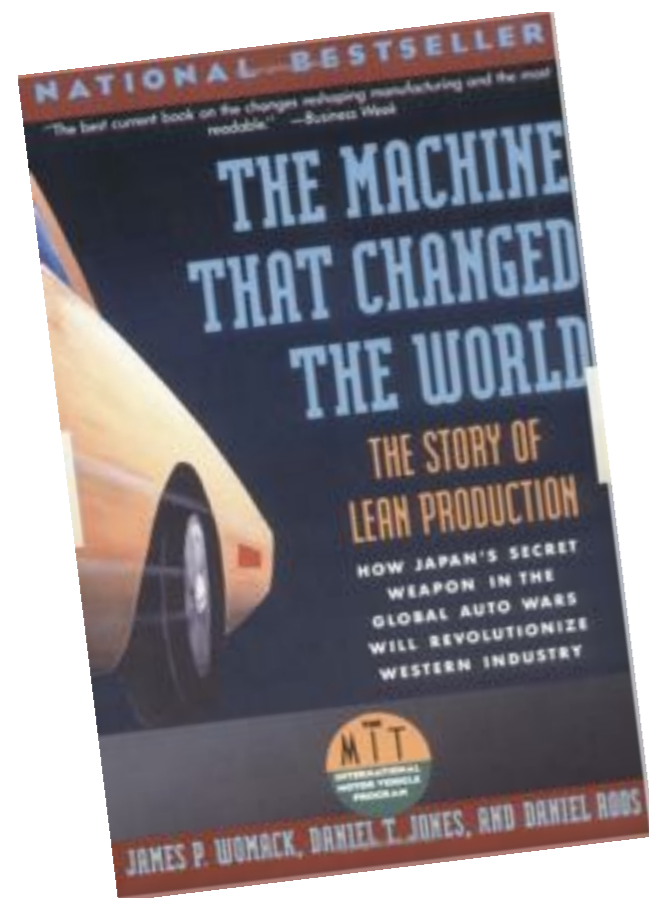
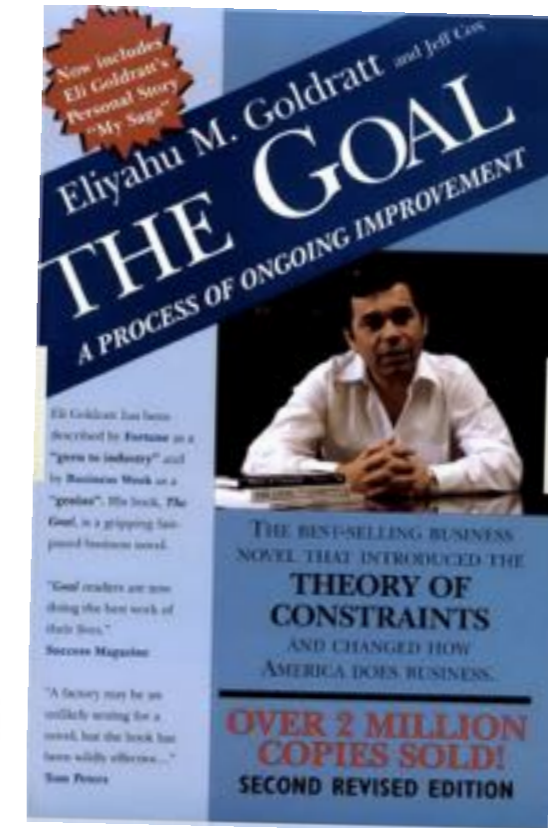
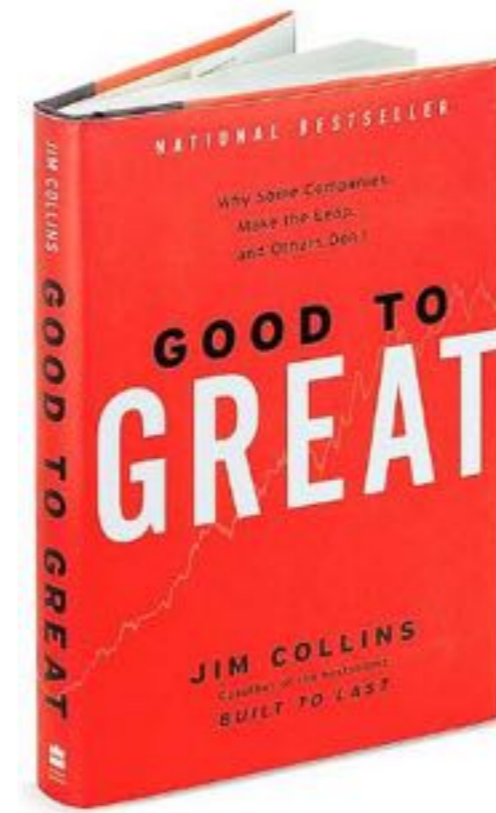
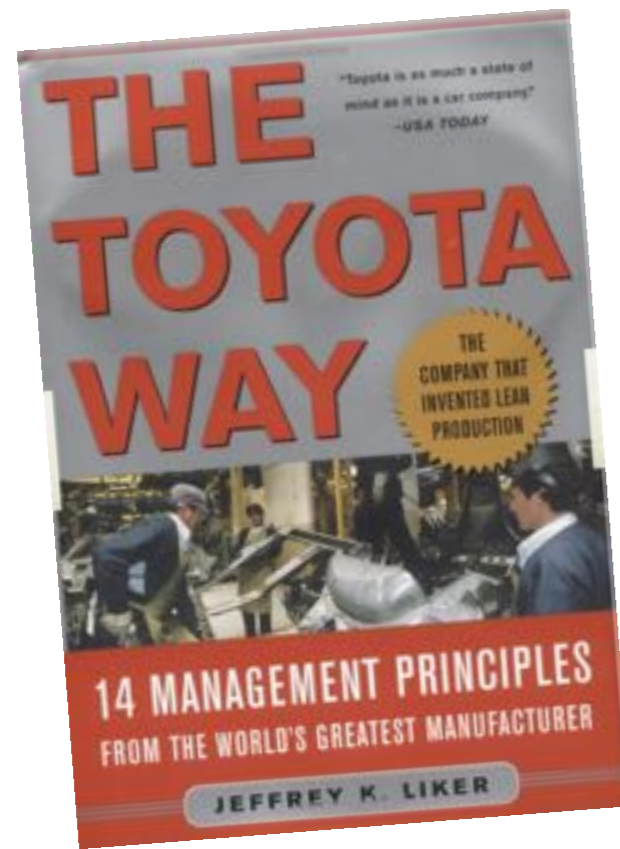


Unfortunately, by the time Rupert's mother had finished knitting his outfit, the war was over.

# Effective Product Development

- Embrace chaos
- Break the rules
- Respect doers
- Focus on communication
- Introduce slack
- Beware the observer effect
- Constrain innovation
- Reward courage
- Focus on the whole product
- Delay decisions
- Aim for approximately right
- Release early, release often
- Follow principles, not processes
- Timing is everything







Cisco Systems, Innovation Center  
Lysaker, Norway



det kan noen ganger se slik ut...

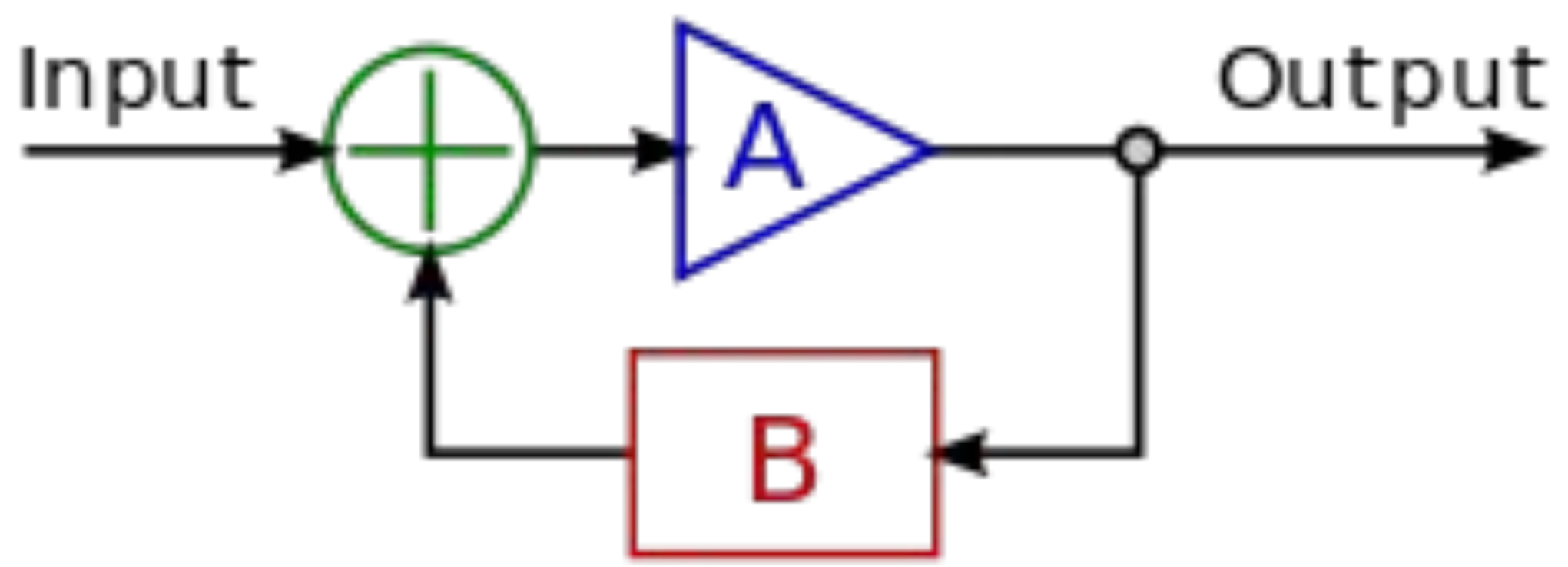


men vi jobber nesten alltid i team...



typisk er vi 10-60 personer som jobber med prosjekter.

!



# Focus on flow



# Control does not always work





The more you tighten your grip, Tarkin, the more  
star systems will slip through your fingers.

(Princess Leia)

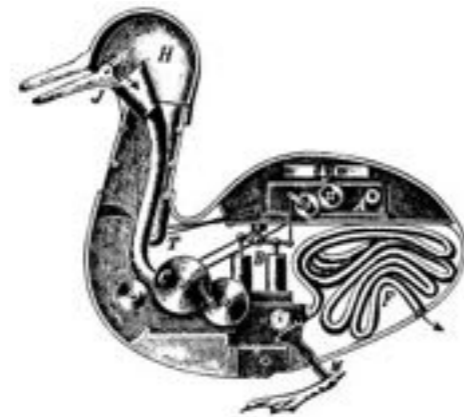


Make sure that everybody is working towards a common goal.



# Reductionism vs Systems thinking

**Reductionism** is a philosophical position that a complex system is nothing but the sum of its parts, and that an account of it can be reduced to accounts of individual constituents.



**Systems thinking** is the process of understanding how things influence one another within a whole



(aka, Taylorism vs Demingism)



Frederick Winslow Taylor (1856-1915)



W. Edwards Deming (1900-1993)



(Demingism)



TopSpeed 25



(Taylorism)

