

The Tandberg Way

olve.maudal@tandberg.com

TANDBERG has never cared much about documentation, procedures, methodologies and risk reduction. However, we do care very much about our culture and our principles. This has enabled us to outperform all competition in the video conferencing and telepresence market during the last decade.

In retrospect, we realize that TANDBERG has for 10–15 years built a culture that is quite compatible with Agile and Lean ideas.

This talk will give a glimpse into how we do product development in TANDBERG R&D at Lysaker. I will show an example of how we developed a particular product with emphasis on software development, before I dive into the principles that we follow.

June 21, 2010

Disclaimer: This is an extremely subjective view of how we do product development in TANDBERG. Please do not assume that it is possible to generalize well over the examples given.

Outline

- About me, Tandberg and what we do
- Product Development in Tandberg
- Observations from Tandberg R&D
- A case study - Advanced Feedback Driven Development
- Some final thoughts about software development

LET ME EXPLAIN
WHAT VIDEO
COMPRESSION IS...

WOULD YOU STOP
IF I POINTED OUT THAT
EVERYONE IN THIS ROOM
EXCEPT YOU IS AN ELECTRICAL
ENGINEER?

ZEROS ARE ROUND AND FAT
COMPARED TO ONES...

I'M BEGGING
YOU...

12/1/04 © 2004 Scott Adams, Inc./Dist. by UFS, Inc.



About me

1992-1995 BEng, Software Engineering, UMIST, Manchester

1995-1996 MSc, Intelligent Robotics, Dept of Artificial Intelligence, Edinburgh

1996-1996 Postgrad, Data Mining and Knowledge Discovery, NTNU, Trondheim

1996-2000 Schlumberger, developing systems for finding oil

2000-2004 BBS, developing systems for electronically moving money

2004-now TANDBERG, developing systems for effective communication between people

Active member of the vibrant geek community in Oslo. Eg, JavaPils, Smidig, JavaZone, XP Meetup, Cantara, Lean Meetup, Rubberducks and Oslo C++ Users Group, and a lot of other things. Also an active member of ACCU.

Blogs regularly on <http://olvemaudal.wordpress.com/> and Twitter @olvemaudal

About TANDBERG

TANDBERG is the leading provider of telepresence, high-definition video conferencing and mobile video products and services. TANDBERG designs, develops and markets systems and software for video, voice and data. The company provides sales, support and value-added services in more than 90 countries worldwide.



TANDBERG shipped its first product, a picture telephone for ISDN in 1993. Since then TANDBERG has grown from a small startup based in Norway into an international company with ~1700 employees and a revenue of 900 MUSD in 2009. Dual headquarters in New York and Oslo.

Around 500 engineers work in R&D with product development, ~300 of them are based at Lysaker in Norway, but we now also have R&D centres in Langley (UK), Ruscombe (UK), Bangalore (IN) and Hamilton (NZ).

www.tandberg.com

About TANDBERG, now part of



TANDBERG is the leading provider of telepresence, high-definition video conferencing and mobile video products and services. TANDBERG designs, develops and markets systems and software for video, voice and data. The company provides sales, support and value-added services in more than 90 countries worldwide.

TANDBERG shipped its first product, a picture telephone for ISDN in 1993. Since then TANDBERG has grown from a small startup based in Norway into an international company with ~1700 employees and a revenue of 900 MUSD in 2009. Dual headquarters in New York and Oslo.

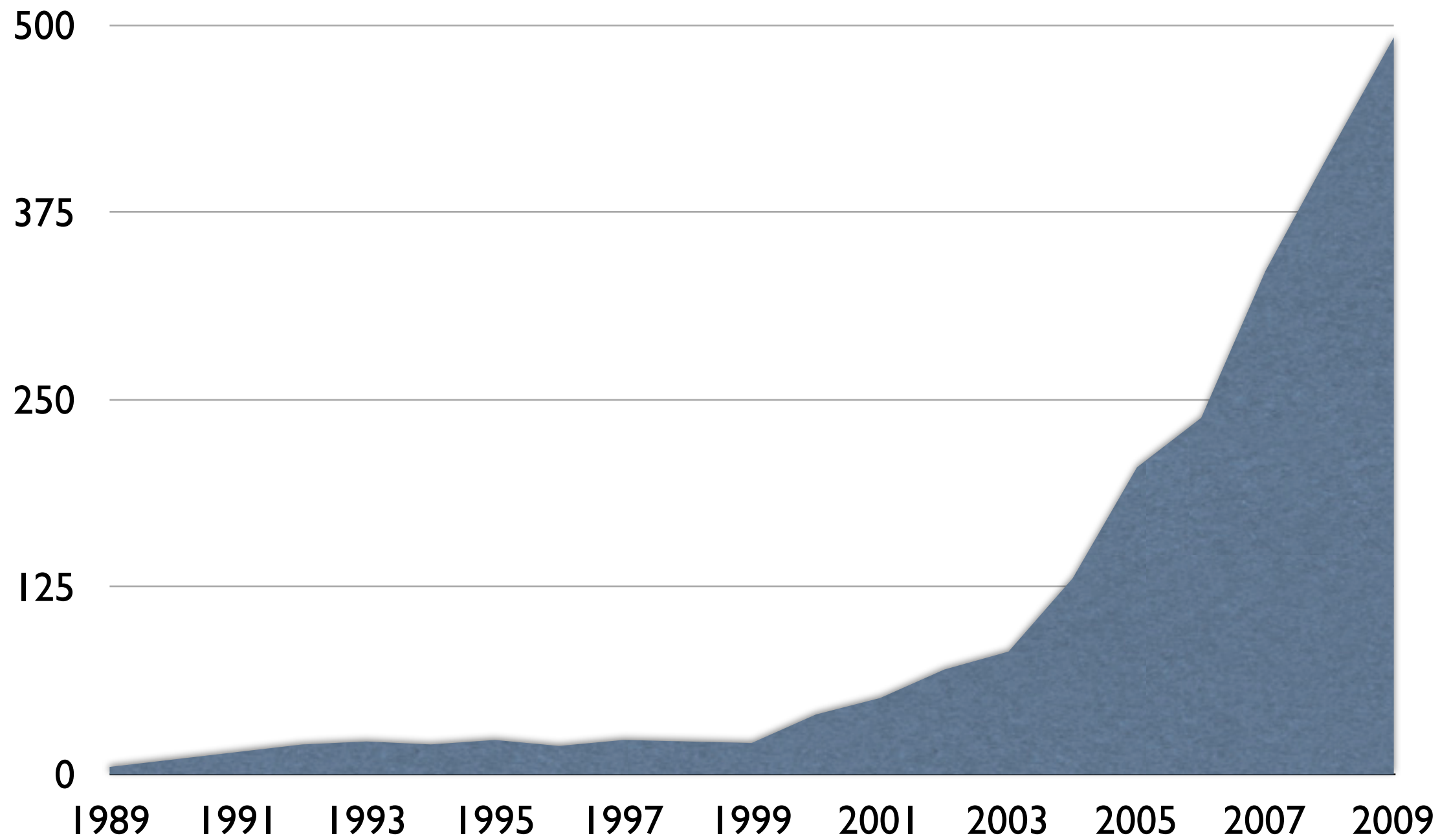
Around ~~500~~⁷⁵⁰ engineers work in R&D with product development, ~300 of them are based at Lysaker in Norway, but we now also have R&D centres in Langley (UK), Ruscombe (UK), Bangalore (IN) and Hamilton (NZ). **And San Jose (US)**

www.tandberg.com

Breaking news (April 18, 2010):

After a successful \$3.8 billion deal, TANDBERG is now part of Cisco

■ Number of employees in TANDBERG R&D



(the numbers are not exact)



Lysaker, PP22, Norway



We develop and sell...



Meeting room systems



Telepresence systems



Personal systems



PC based solutions



Networking products



And a lot of other stuff



~1700 employees worldwide

~500 R&D engineers

~300 write code every day

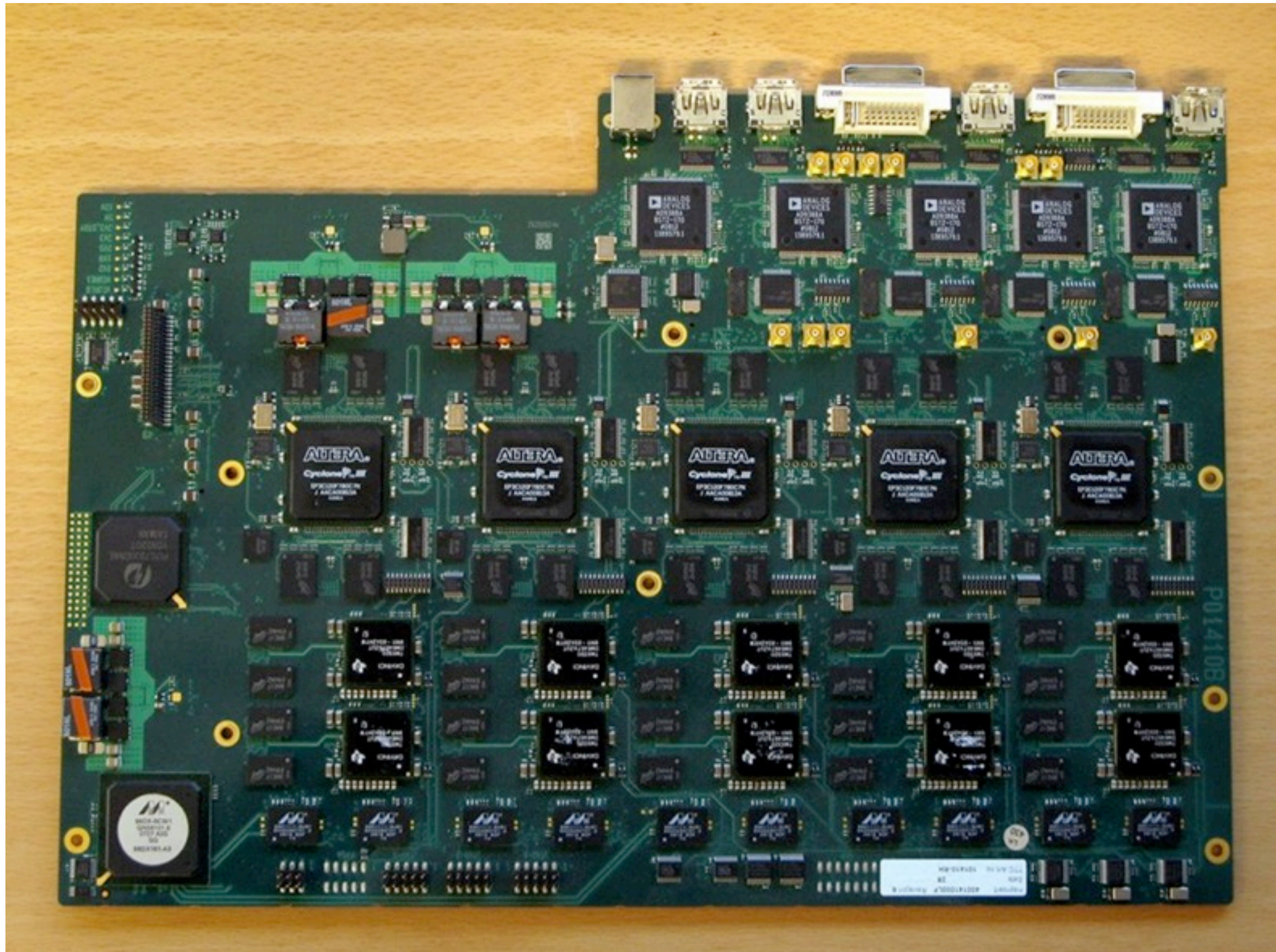
C, C++, Python, Java, C#, VHDL, Ruby

most of us work with software development

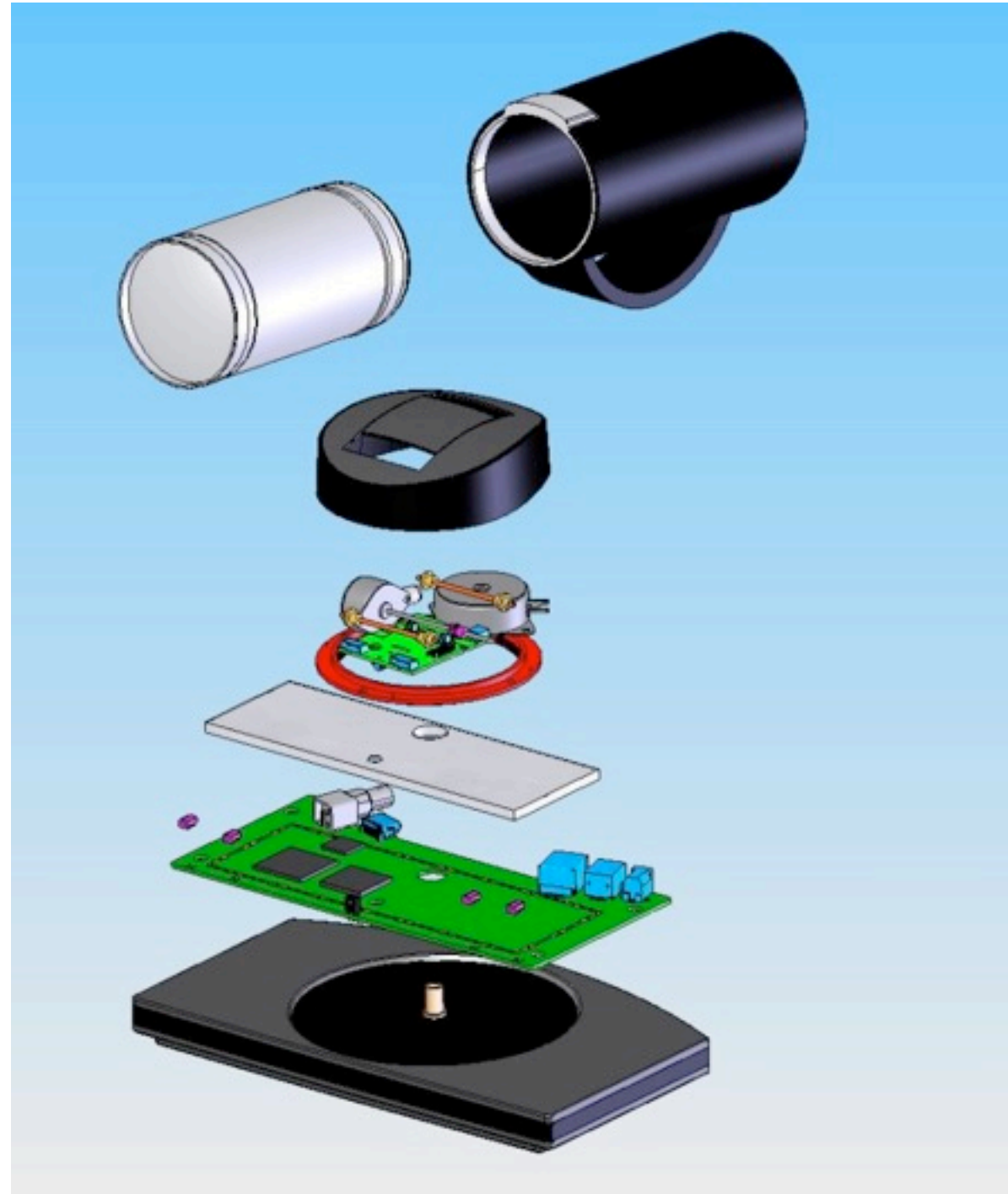


but we also do...

Electronics / Hardware



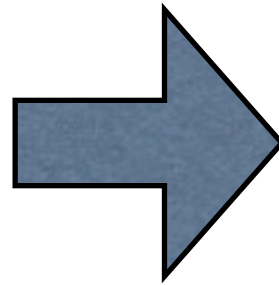
Mechanics



Industrial Design



1992



2007

Looking into



the future



March 2010

“For most of my life I wanted a Porsche, now I think I want a Tandberg EX90 instead.” Wainhouse Research



Video: The new way of working (2:00)

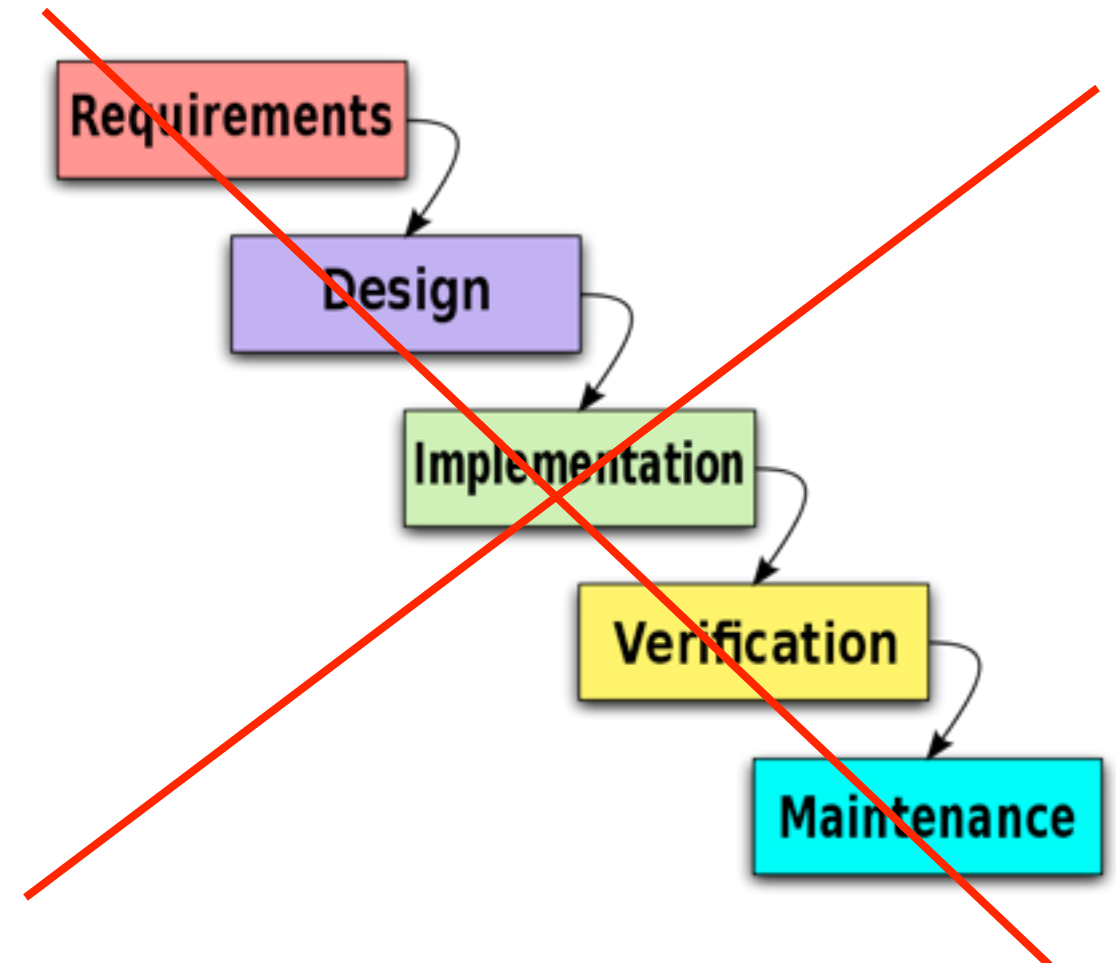
<http://www.tandberg.com/media/index.jsp?id=1373>



Product Development in TANDBERG

- No corporate standards or procedures
- Little documentation gives effective communication
- Treat engineers as professionals, not as resources
- Slack is embedded, and “skunk work” projects appreciated
- "Plans are nothing, planning is everything"
- No time recording, and we do not measure project cost
- To fail is OK, therefore we deliver spectacular stuff
- Doers are very much respected in Tandberg
- Autonomous organization
- Communication is a key skill for all our engineers
- We are fast and “sloppy”
- We release early and we release often
- Fun gives profit (not: profit, then fun)
- The company builds on trust

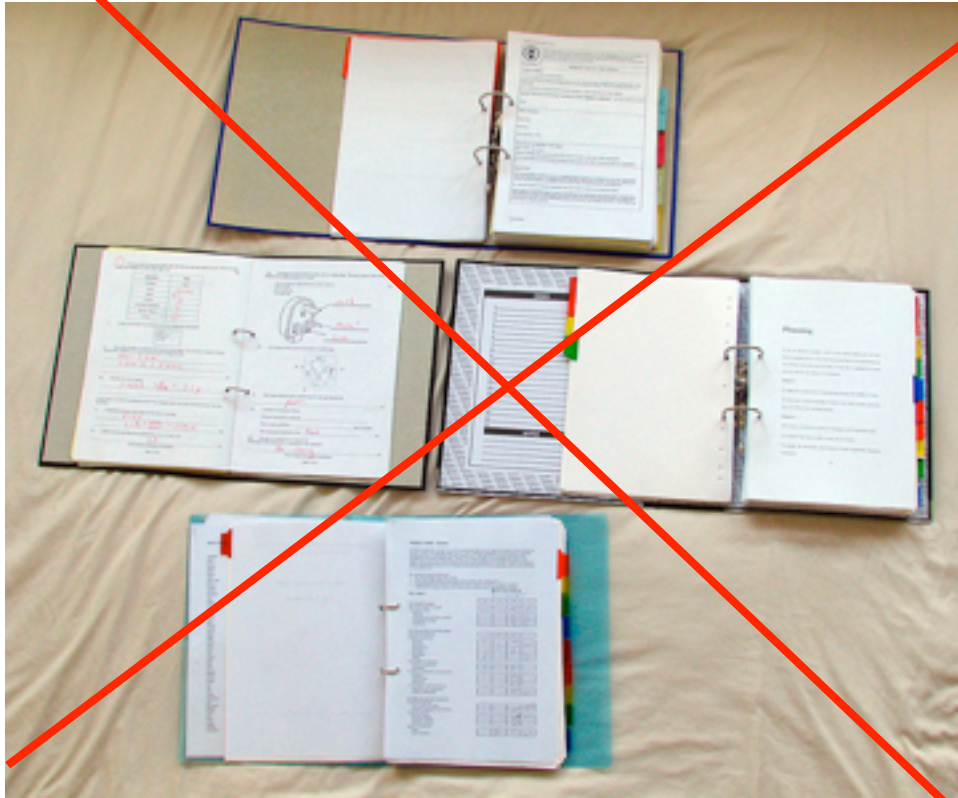
No corporate standards or procedures



[...] we have been designing complex systems whose active components are variable and highly non-linear components called people, without characterizing these components or their effect on the system being designed. Upon reflection, this seems absurd [...]

(Alistair Cockburn, 1999, about "traditional" software methodologies)

Little documentation gives effective communication



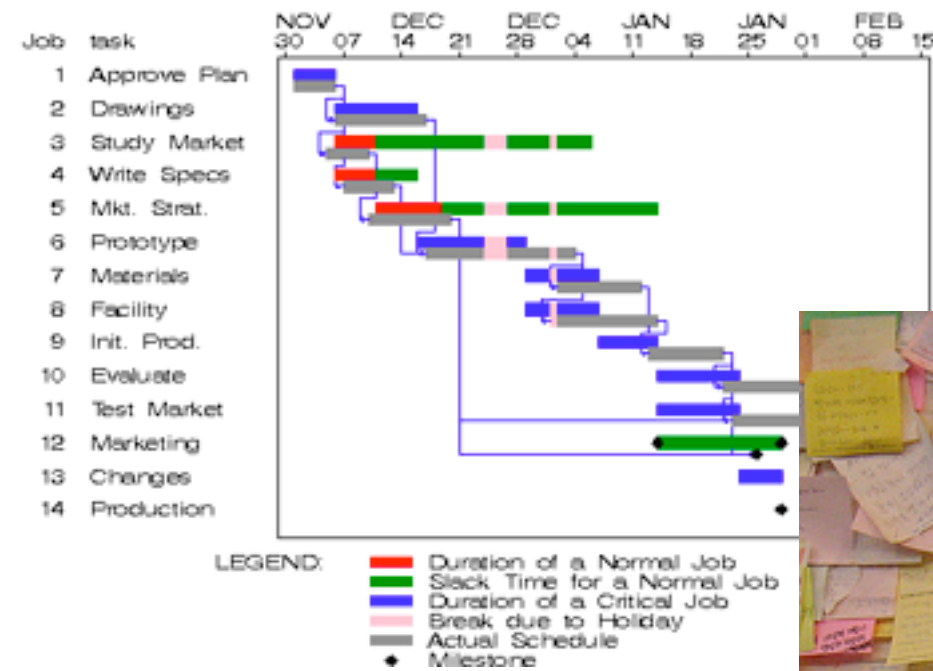
Treat engineers as professionals, not as resources



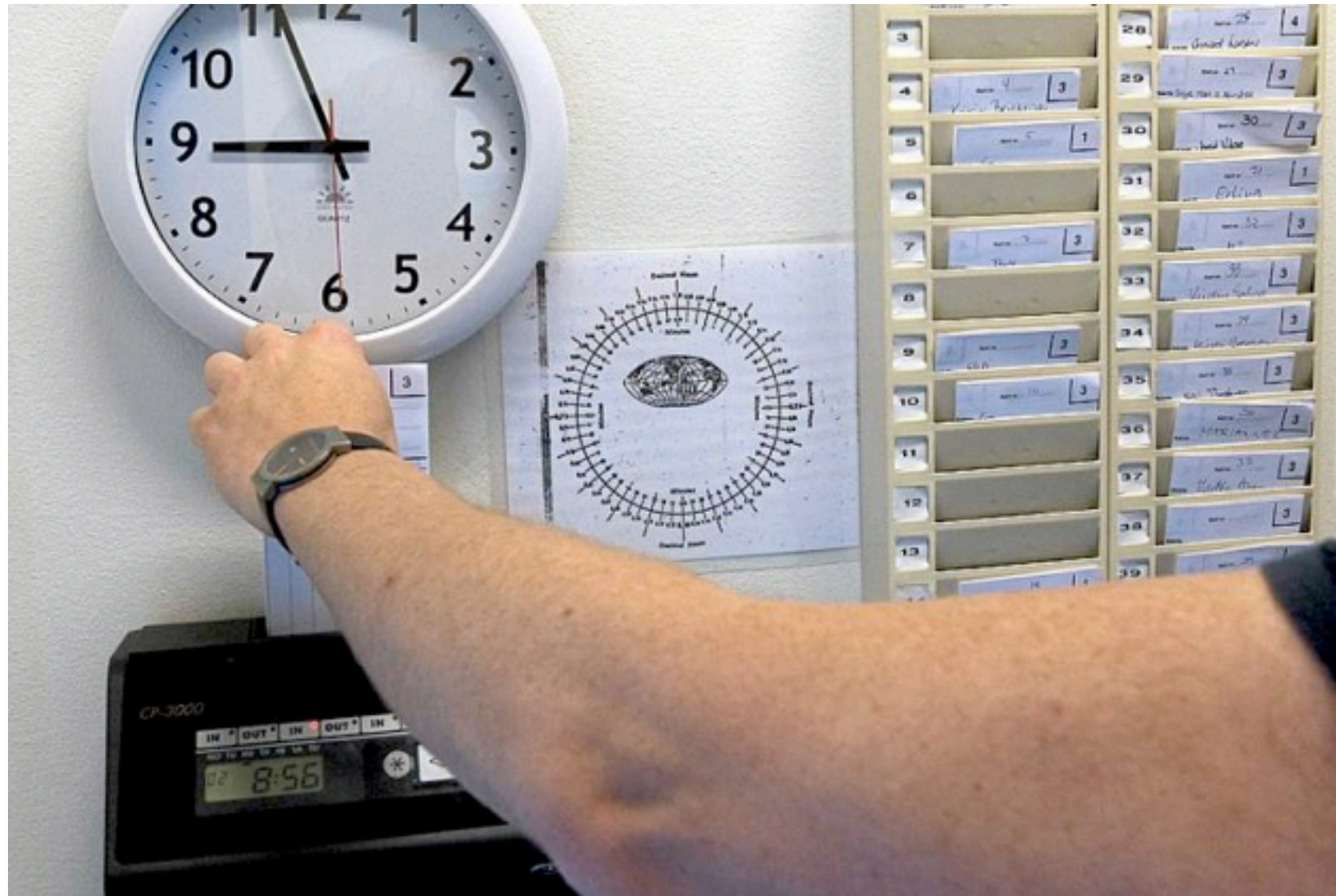
Slack is embedded, and “skunk work” projects appreciated



"Plans are nothing, planning is everything"



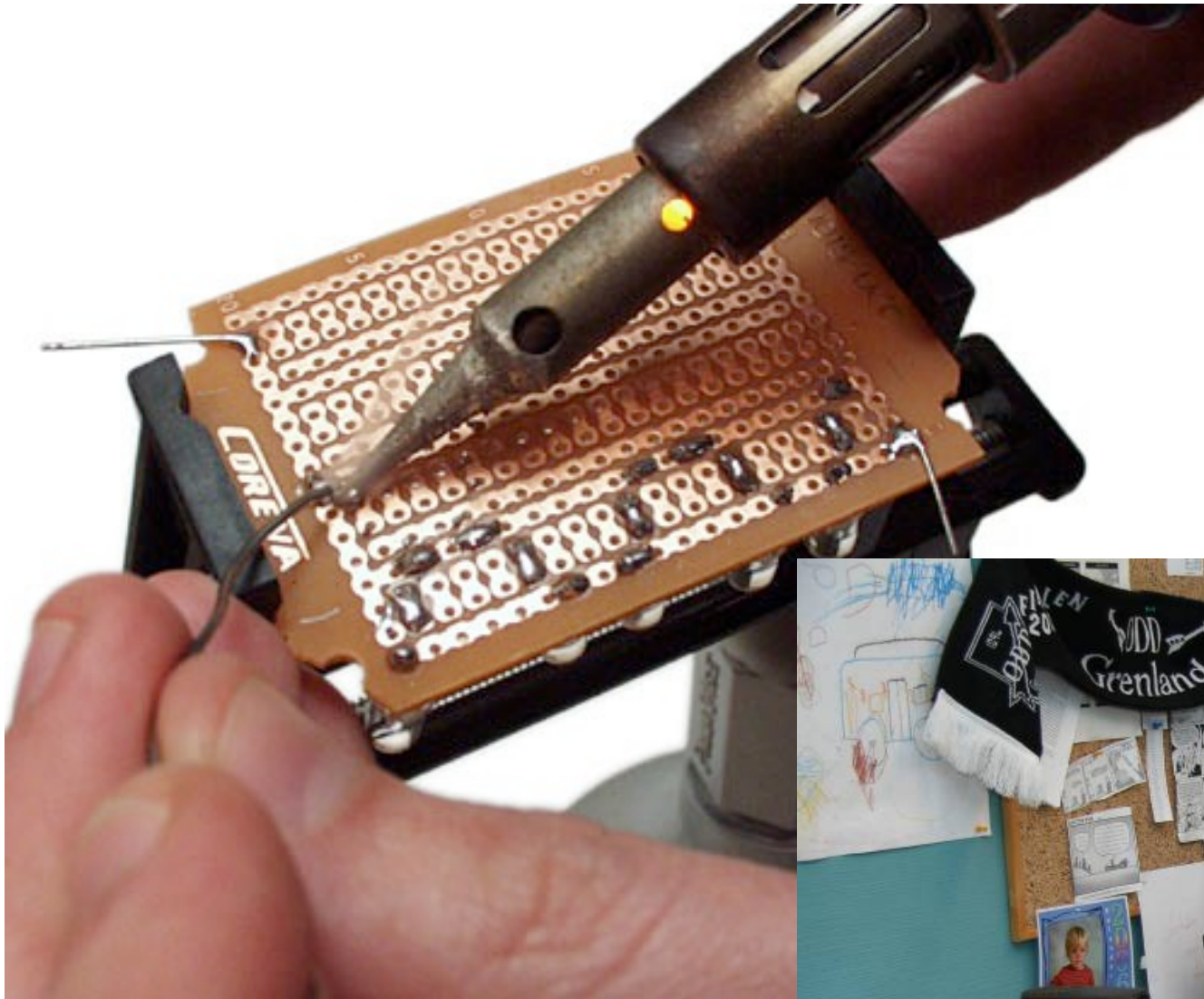
No time recording, and we do not measure project cost



To fail is OK, therefore we deliver spectacular stuff



Doers are very much respected in Tandberg



Autonomous organization



Communication is a key skill for all our engineers



We are fast and “sloppy”

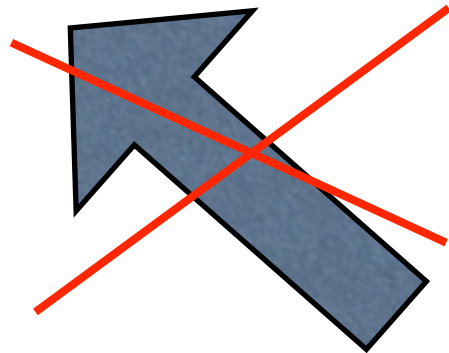
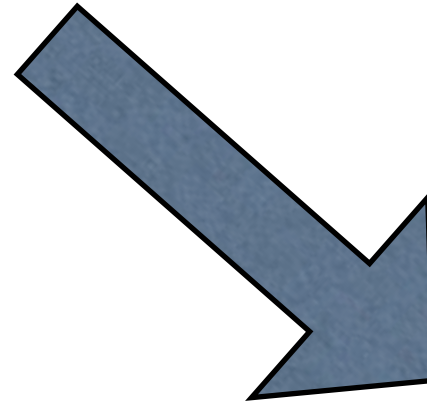


Itzhak Perlman plays Bazzini

We release early and we release often



Fun gives profit (not: profit, then fun)



The company builds on trust



Product Development in TANDBERG

- No corporate standards or procedures
- Little documentation gives effective communication
- Treat engineers as professionals, not as resources
- Slack is embedded, and “skunk work” projects appreciated
- "Plans are nothing, planning is everything"
- No time recording, and we do not measure project cost
- To fail is OK, therefore we deliver spectacular stuff
- Doers are very much respected in Tandberg
- Autonomous organization
- Communication is a key skill for all our engineers
- We are fast and “sloppy”
- We release early and we release often
- Fun gives profit (not: profit, then fun)
- The company builds on trust

We follow principles, not processes!

Observations from TANDBERG

As a software engineer joining TANDBERG...

at first you might get this impression...

- No documentation
- No routines
- Fooling around
- Not following plans
- Decision are postponed
- Nobody decides
- Little respect for management
- Little modularization
- Lack of precision
- Sloppiness
- People are not working hard

but then you start to notice...

- No documentation
- No routines
- Fooling around
- Not following plans
- Decision are postponed
- Nobody decides
- Little respect for management
- Little modularization
- Lack of precision
- Sloppiness
- People are not working hard
- People communicate
- Focus on important stuff
- Embedded slack
- Continuous planning
- Effective decisions
- Autonomous organisation
- Respect for the doers
- No integration period
- Spectacular products
- Fast deliveries
- Sustainable pace

... and while you still see the "negative" stuff, you will start to appreciate the "positive" stuff more.

- No documentation
- No routines
- Fooling around
- Not following plans
- Decision are postponed
- Nobody decides
- Little respect for management
- Little modularization
- Lack of precision
- Sloppiness
- People are not working hard

- People communicate
- Focus on important stuff
- Embedded slack
- Continuous planning
- Effective decisions
- Autonomous organisation
- Respect for the doers
- No integration period
- Spectacular products
- Fast deliveries
- Sustainable pace

Observations from TANDBERG

- People communicate
- Focus on important stuff
- Embedded slack
- Continuous planning
- Effective decisions
- Autonomous organisation
- Respect for the doers
- No integration period
- Spectacular products
- Fast deliveries
- Sustainable pace



Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

The 7 Lean Software Development Principles

- Eliminate Waste
- Create Knowledge
- Build Quality In
- Defer Commitment
- Deliver Fast
- Respect People
- Improve the System

(Poppendieck)

Lean Software Development An Agile Toolkit

Now includes
Eli Goldratt's
Personal Story

Goldratt and Jeff Cox
GOAL
IMPROVEMENT

NEED TO KNOW ABOUT
MANUFACTURING
IN JOE'S GARAGE

NATIONAL BEST
SELLER

"The best current book on the changes reshaping manufacturing is readable." —Business Week

THE MAN THAT CHANGED THE WORLD

Ta O

THE TOYOTA WAY

"Toyota is as much a state of
mind as it is a car company."
—USA TODAY

THE
COMPANY THAT
INVENTED LEAN
PRODUCTION



14 MANAGEMENT PRINCIPLES FROM THE WORLD'S GREATEST MANUFACTURER

JEFFREY K. LIKER

TOYOTA PRODUCTION SYSTEM
Beyond Large-Scale

W. EDWARDS
DEMING



OF CRISIS

JAMES P. WOMACK, DANIEL T. JONES, AND DAVID
MAGALHAES

THE

TANDBERG

WAY

A case study:

TANDBERG Codec C90 - “The Saturn Project”



How did we do it?

Disclaimer:

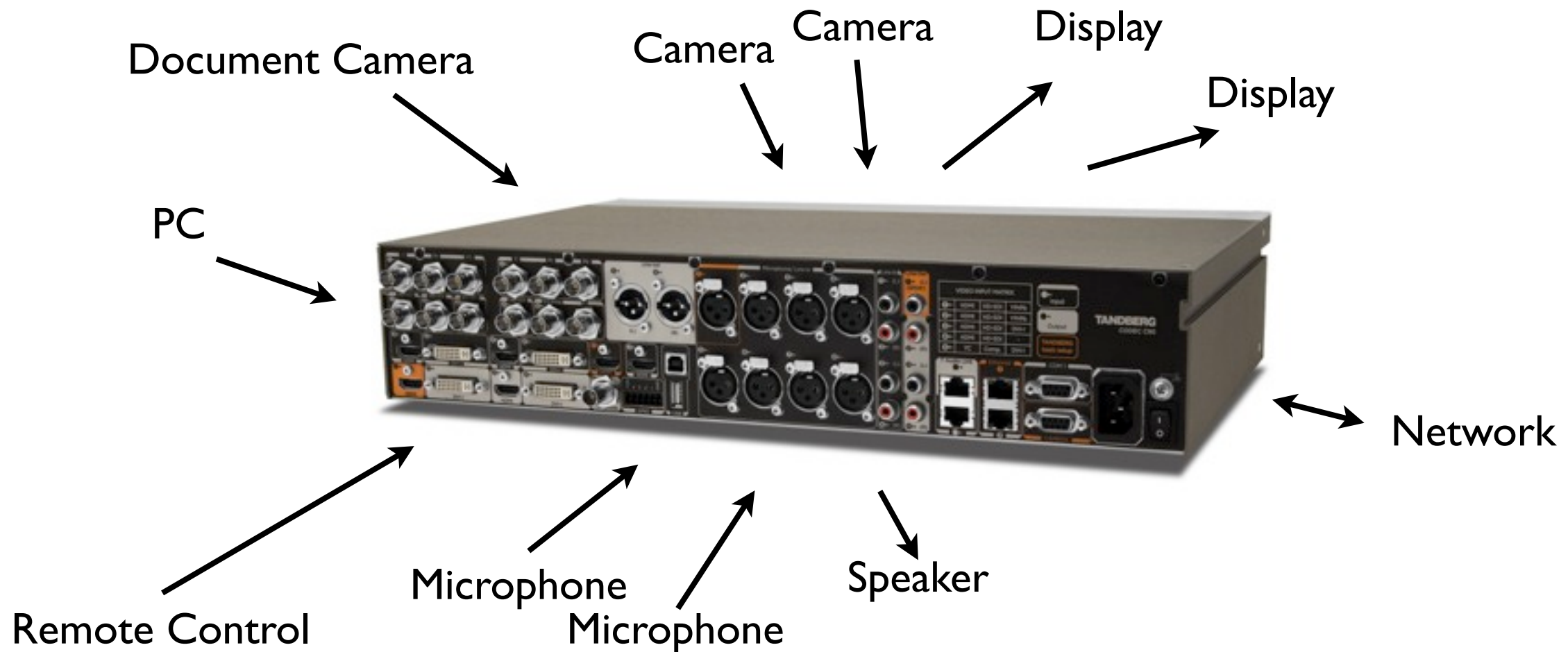
The following description does not show how projects in Tandberg are typically developed, it is just an example of how a particular project actually did it. We think about every project, product and team as something unique, thus it does not make sense to create a particular procedure to follow. You can not make spectacular products following canned procedures.

Indeed, when it comes to product development, TANDBERG is “allergic” to corporate procedures. It is “unthinkable” that anyone outside a project or a team should impose a certain way of doing things, so we can not say “This is the way we do it”, but you may look at a particular project and say “This is the way we did it”.



C90 video (1:19)

<http://www.tandberg.com/media/index.jsp?id=1312>



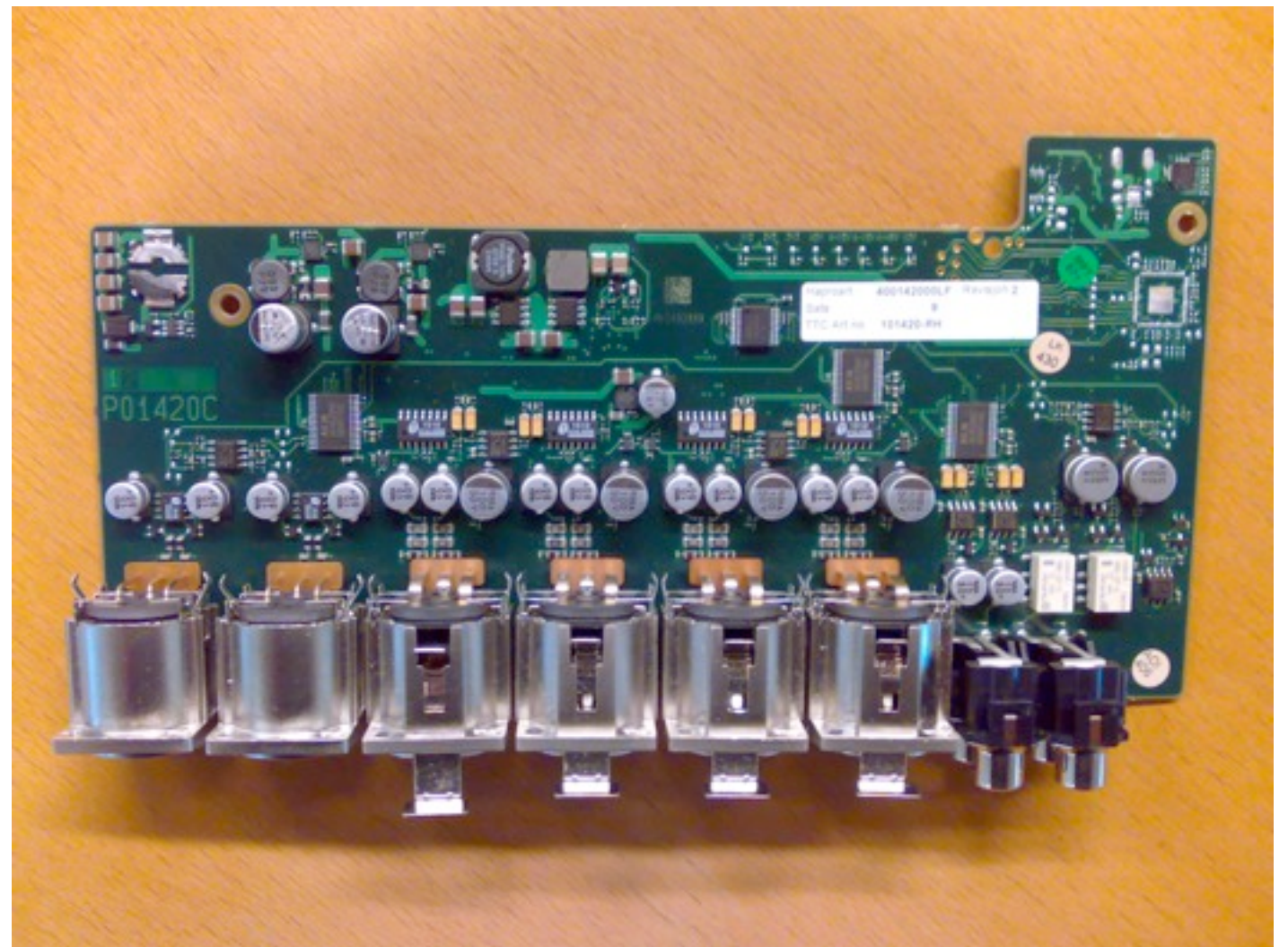
C90 Features:

- realtime H.264 encoding/decoding
- full HD 1080p30, (4+4) concurrent streams
- 12 high definition video sources
- 8 high quality audio sources
- support for many-to-many communication
- Interoperability through H323 and SIP
- API for integration and remote control



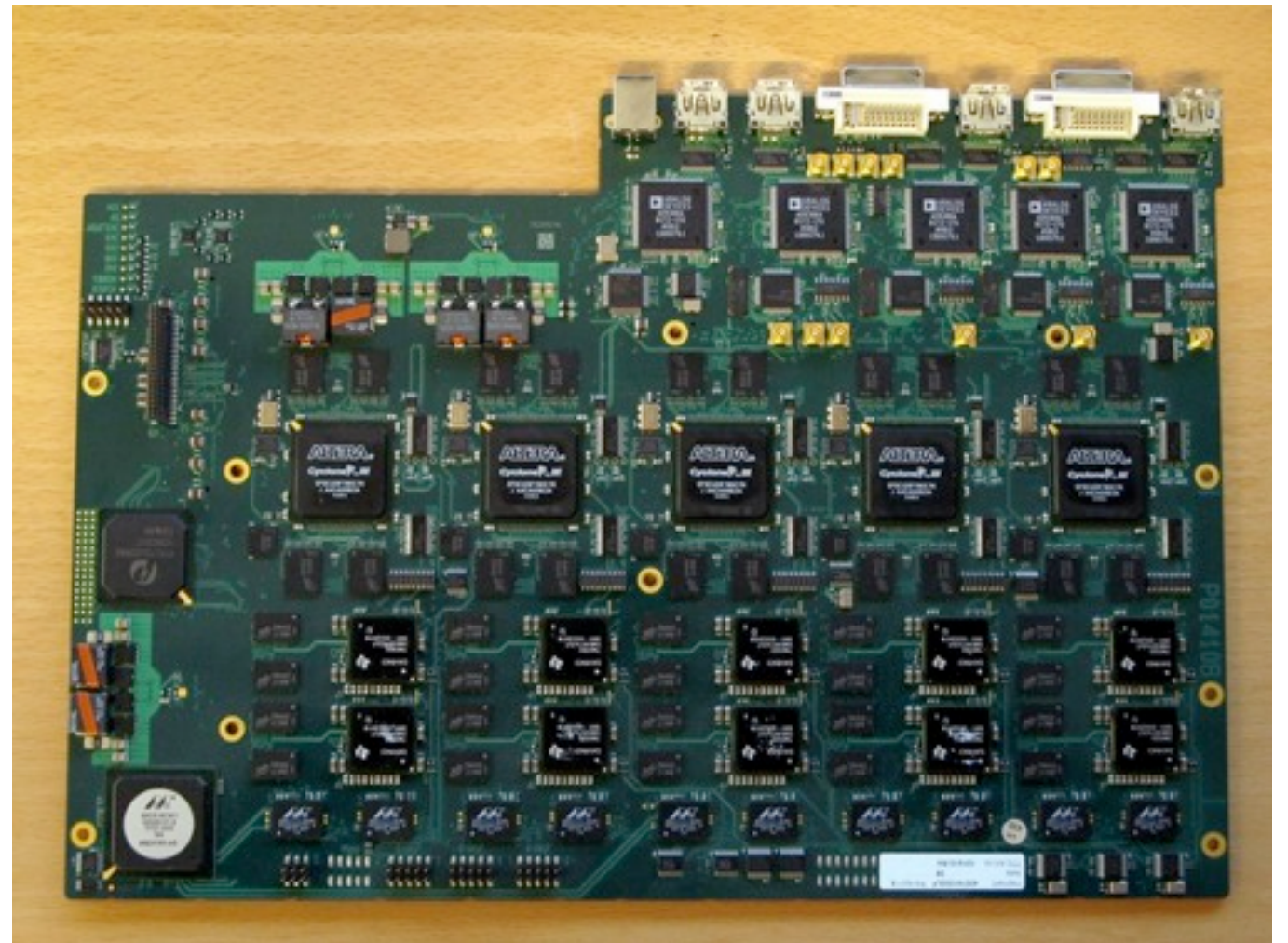
C90 AUDIO EXTENSION BOARD

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



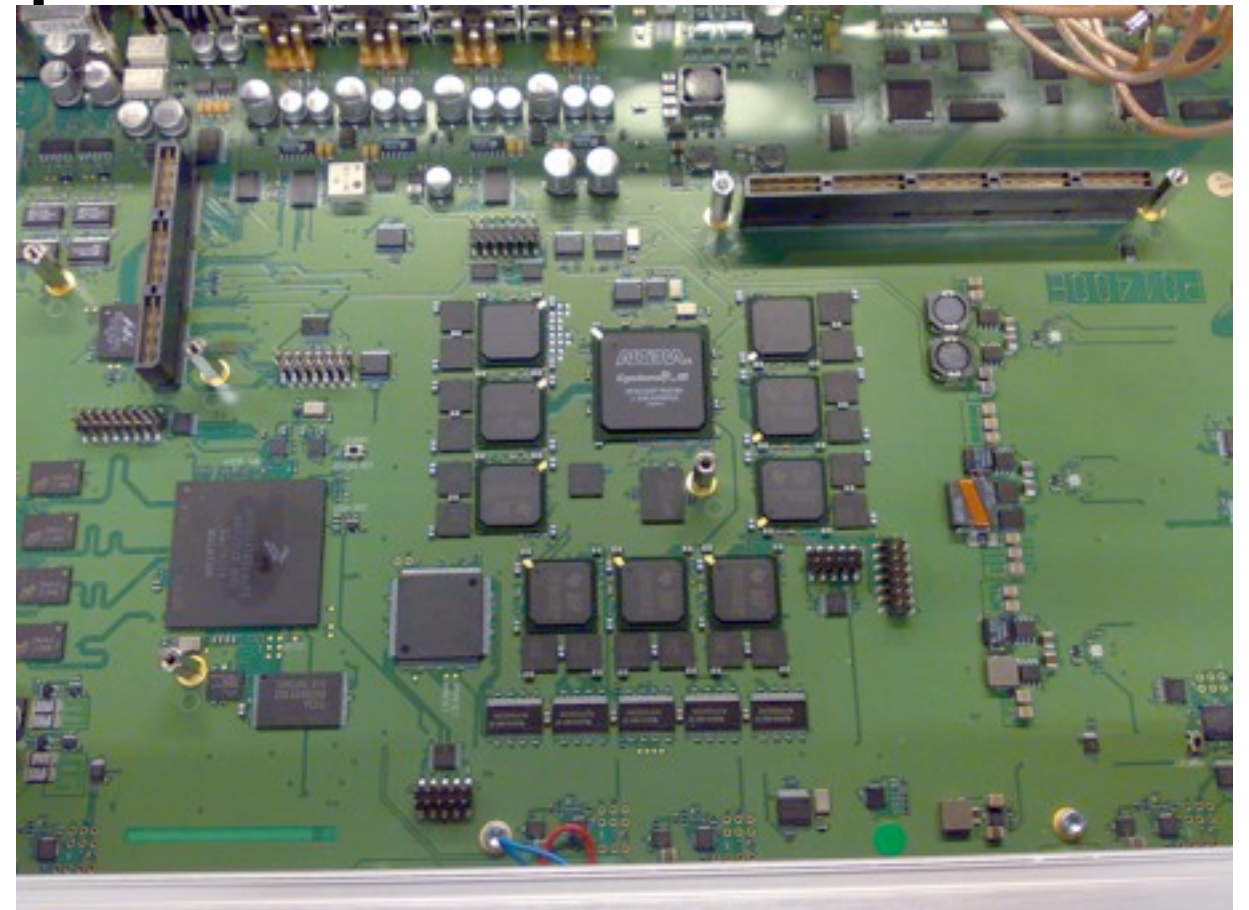
C90 VIDEO BOARD

- 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 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 - from a geek point of view

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

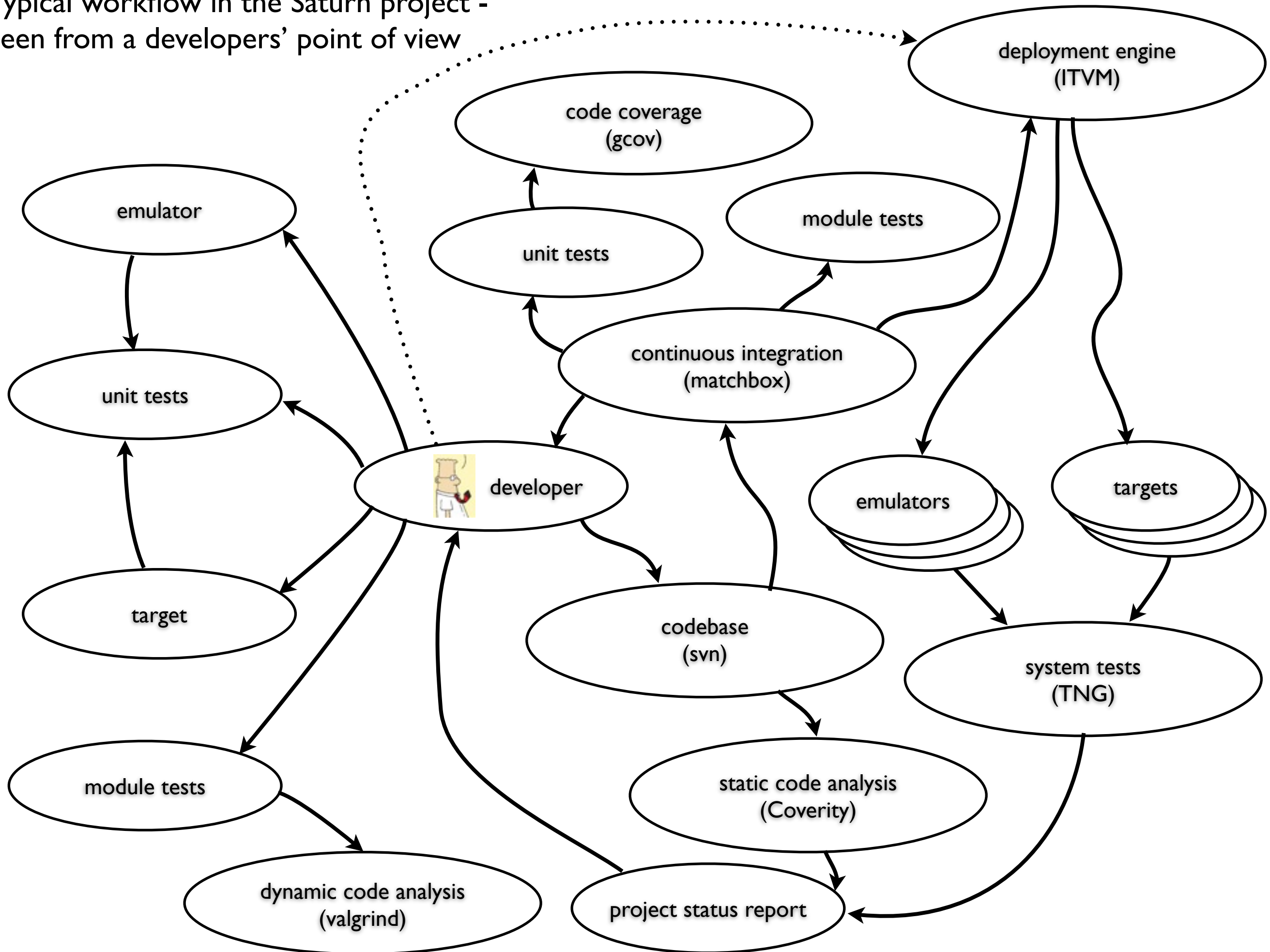
TANDBERG Codec C90

- Developed at Lysaker
- Started spring 2007
- First HW prototype arrived summer 2008
- Released late 2008 (~20 months of development)
- 2-3 people working with mechanics/design
- 4-5 people working with electronics/hardware
- 5-6 people working with FPGA development
- 40-50 people working with software development
- 4 people working with test developers
- 1 person working with approvals

Development Practices in the Saturn project

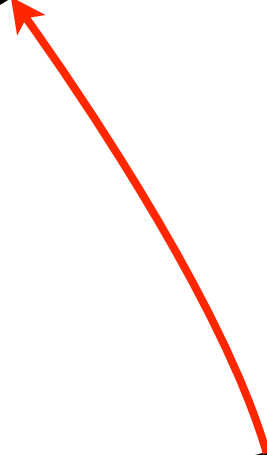
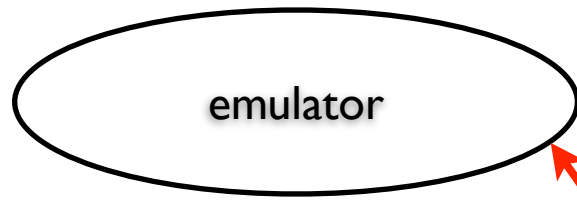
- Continuous planning
- Always attack high risks first
- Heavy focus on effective feedback mechanisms
- Visualization of actual status throughout project
- Teams: GUI, App, Protocol, Video, Audio, FPGA, Platform, QA, Support
- Parallel development
- Iterations and time-boxing
- Daily 15 minute morning assembly of elders
- Weekly rendezvous meetings
- Early and many prototypes

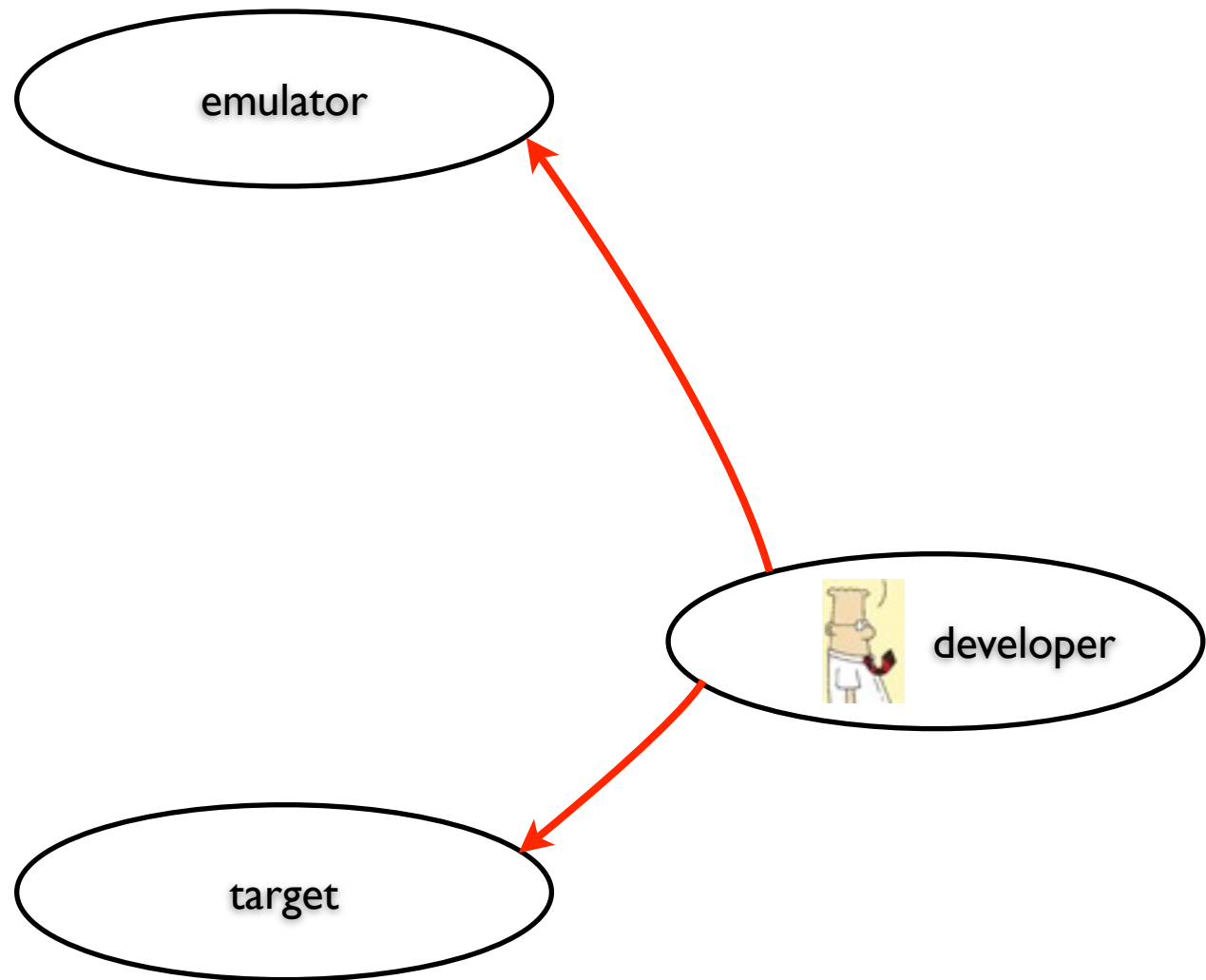
Typical workflow in the Saturn project -
seen from a developers' point of view

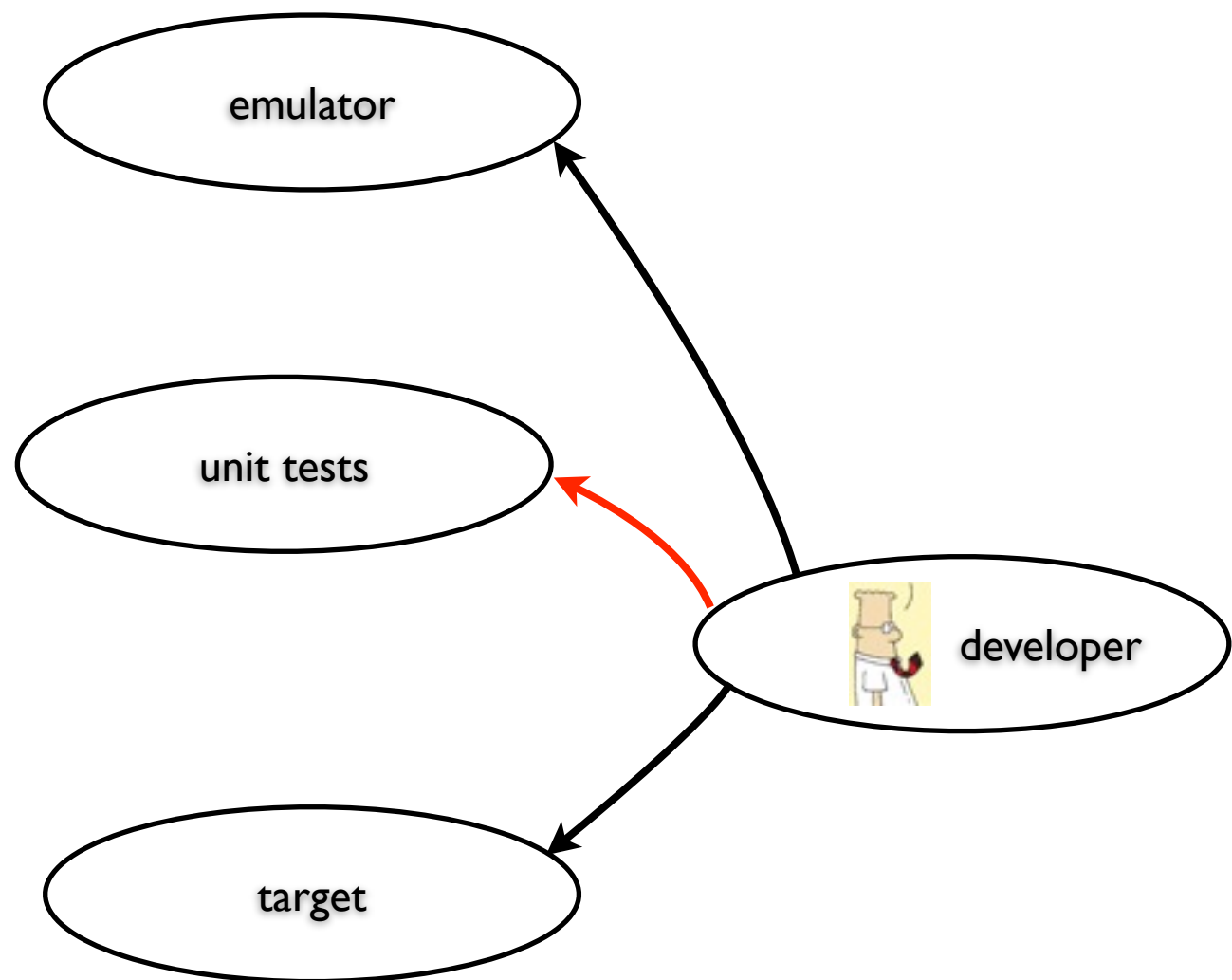


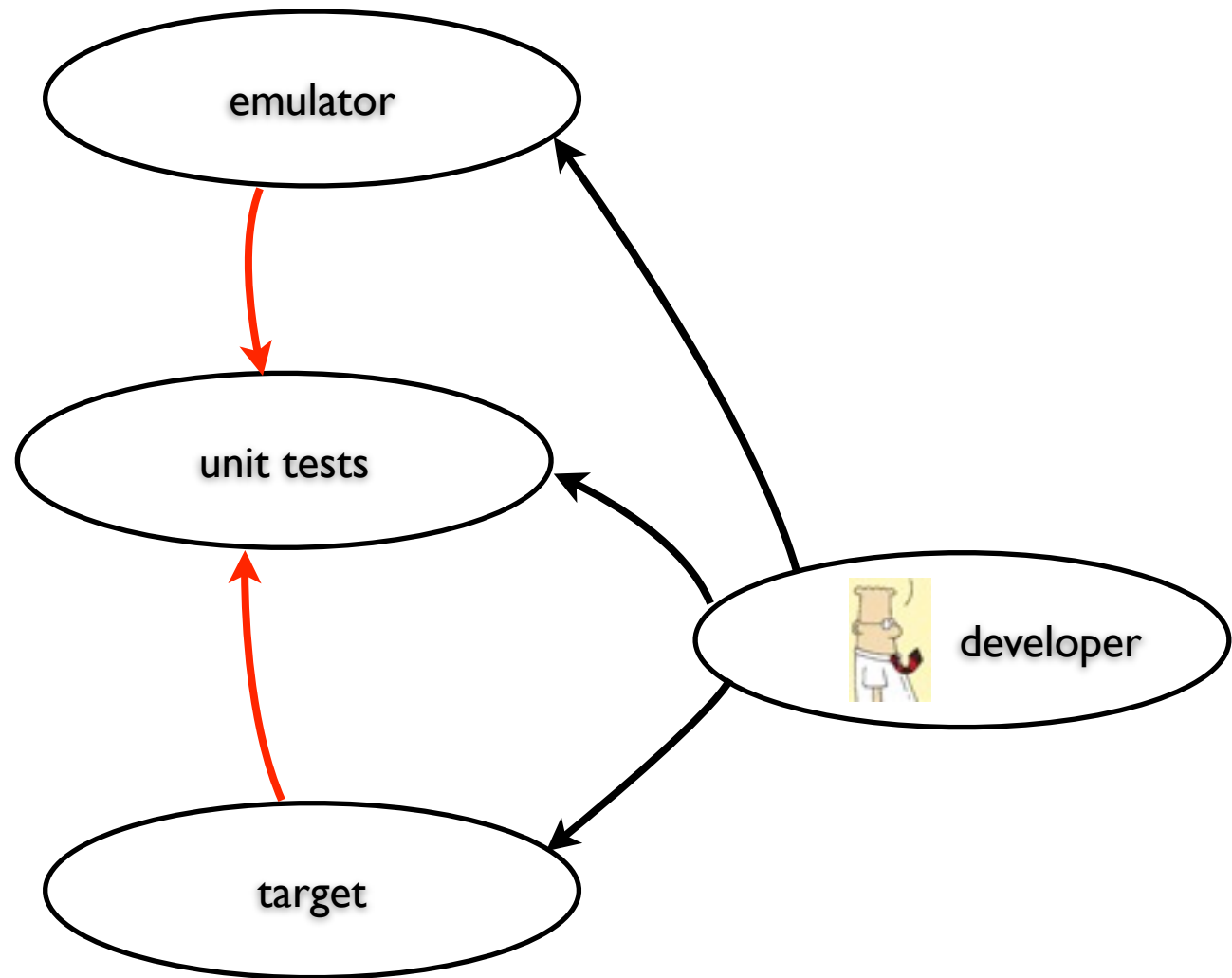
Software development in the Saturn project as
seen from a developers point of view.

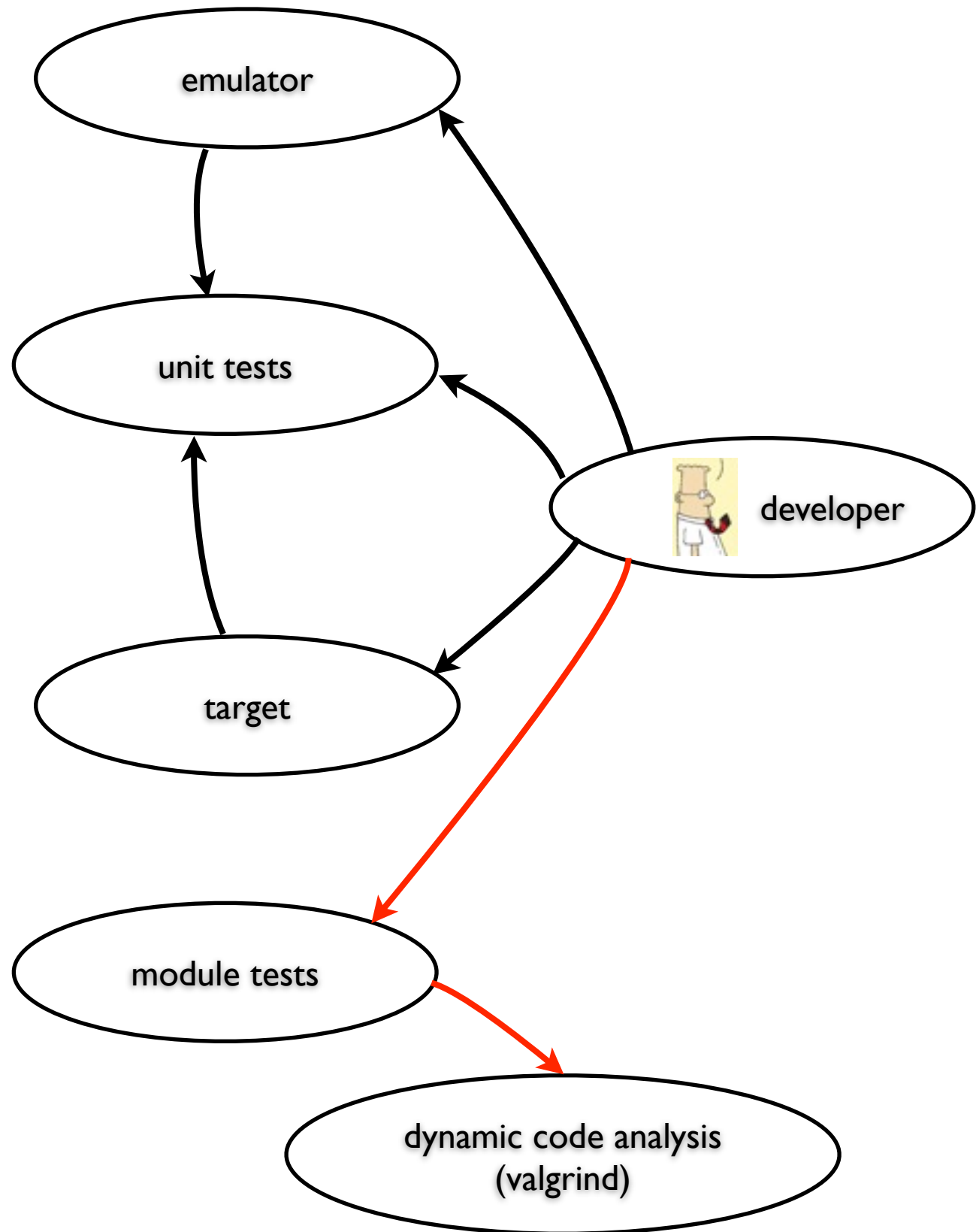


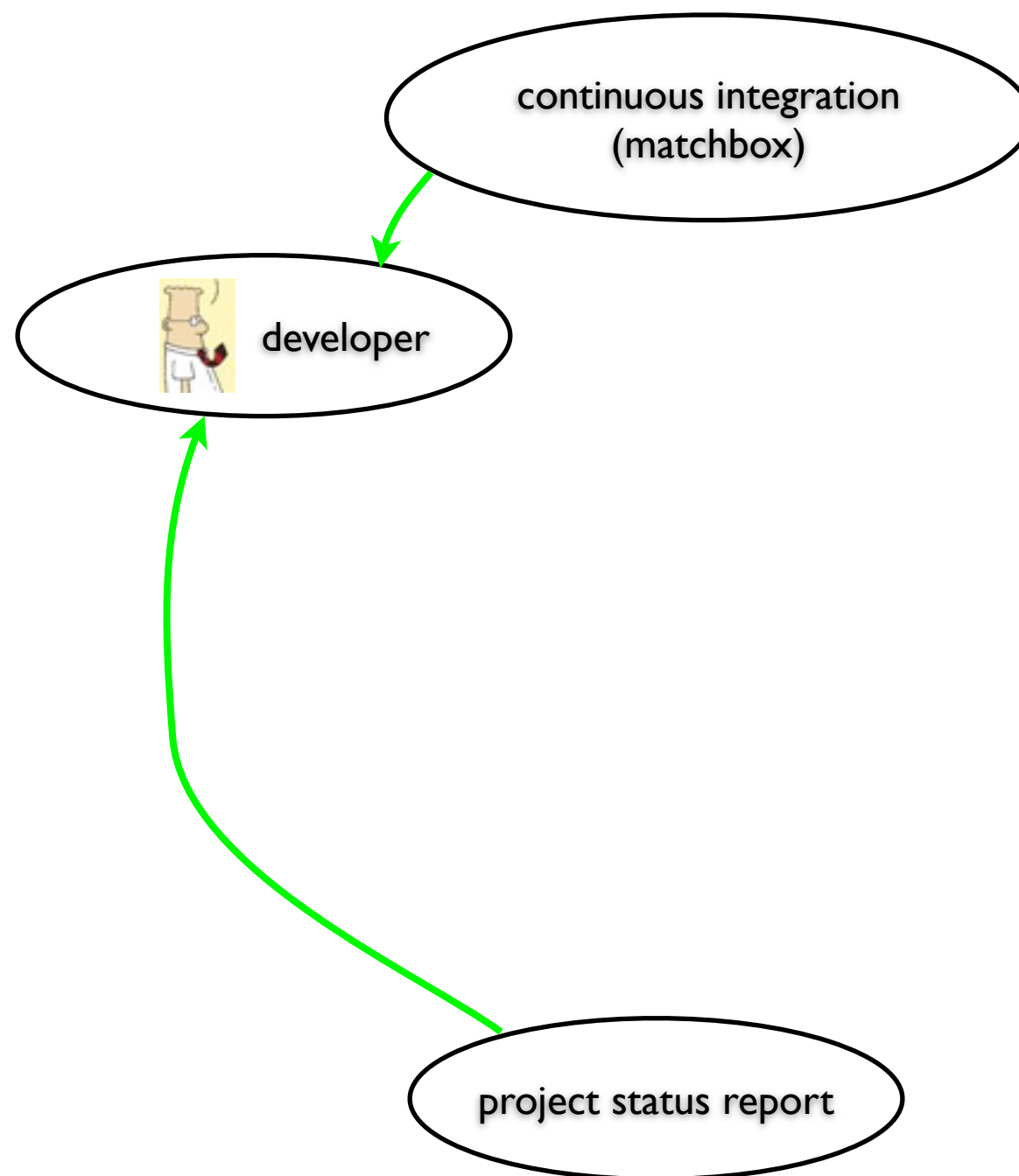


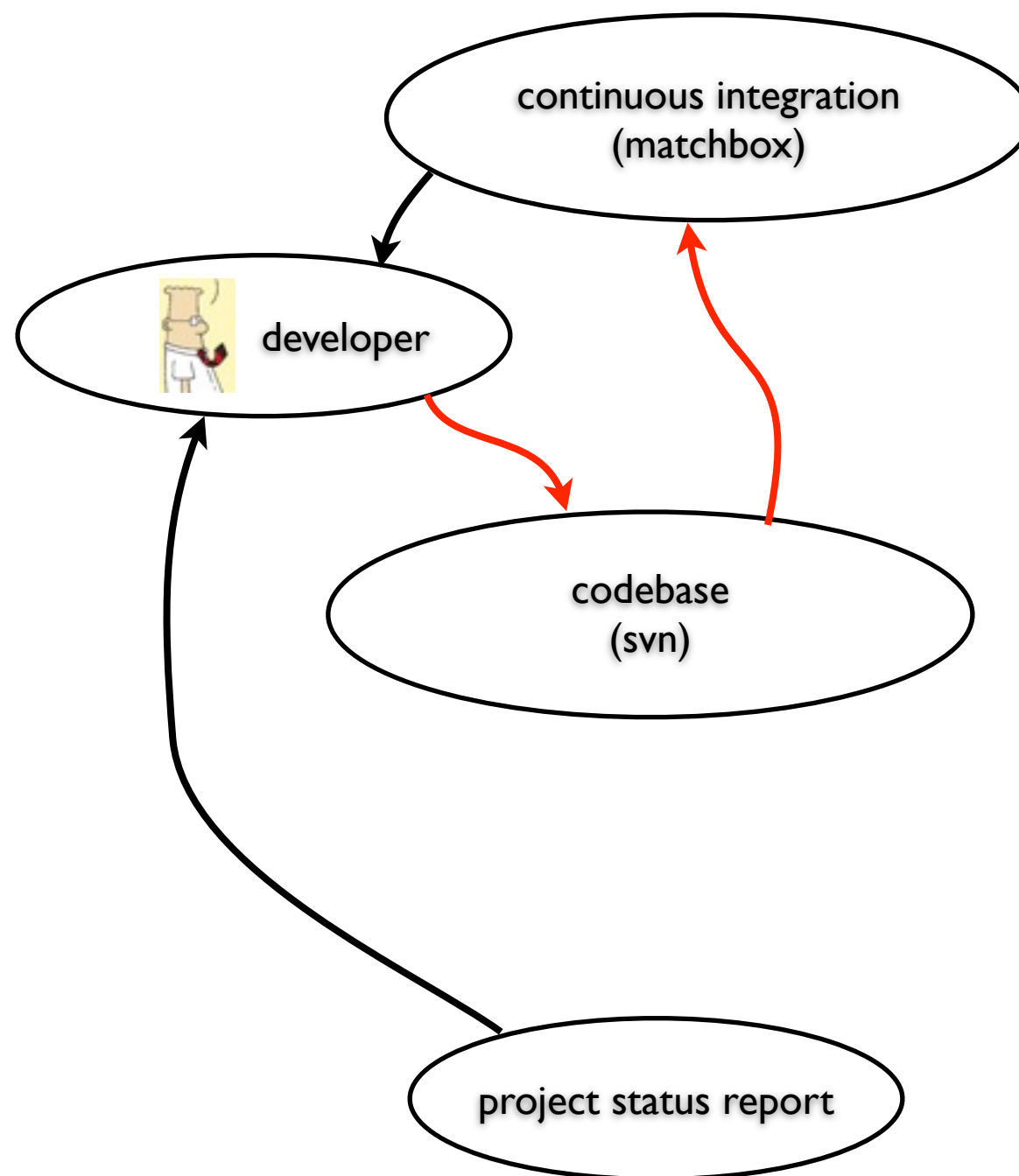


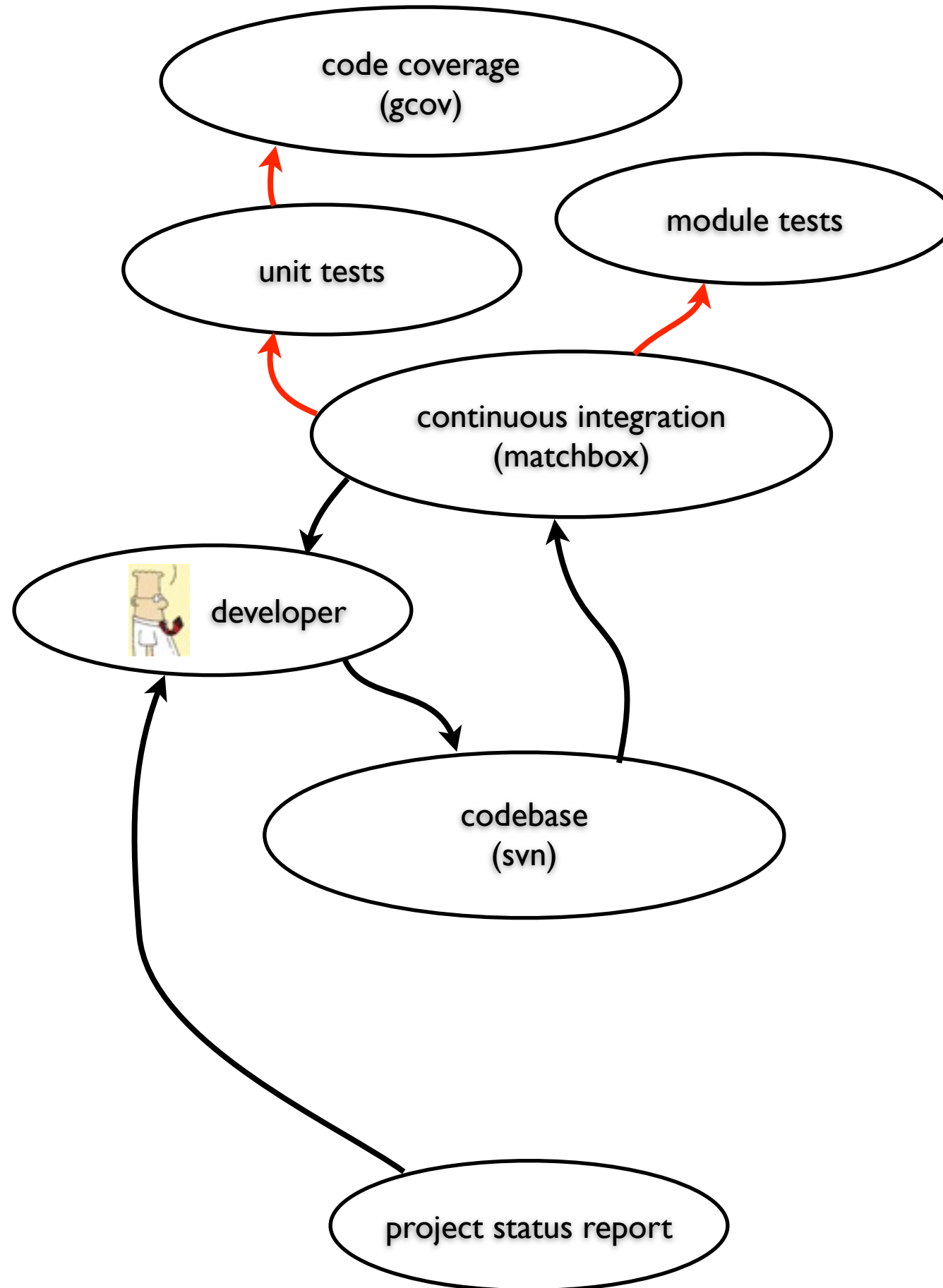


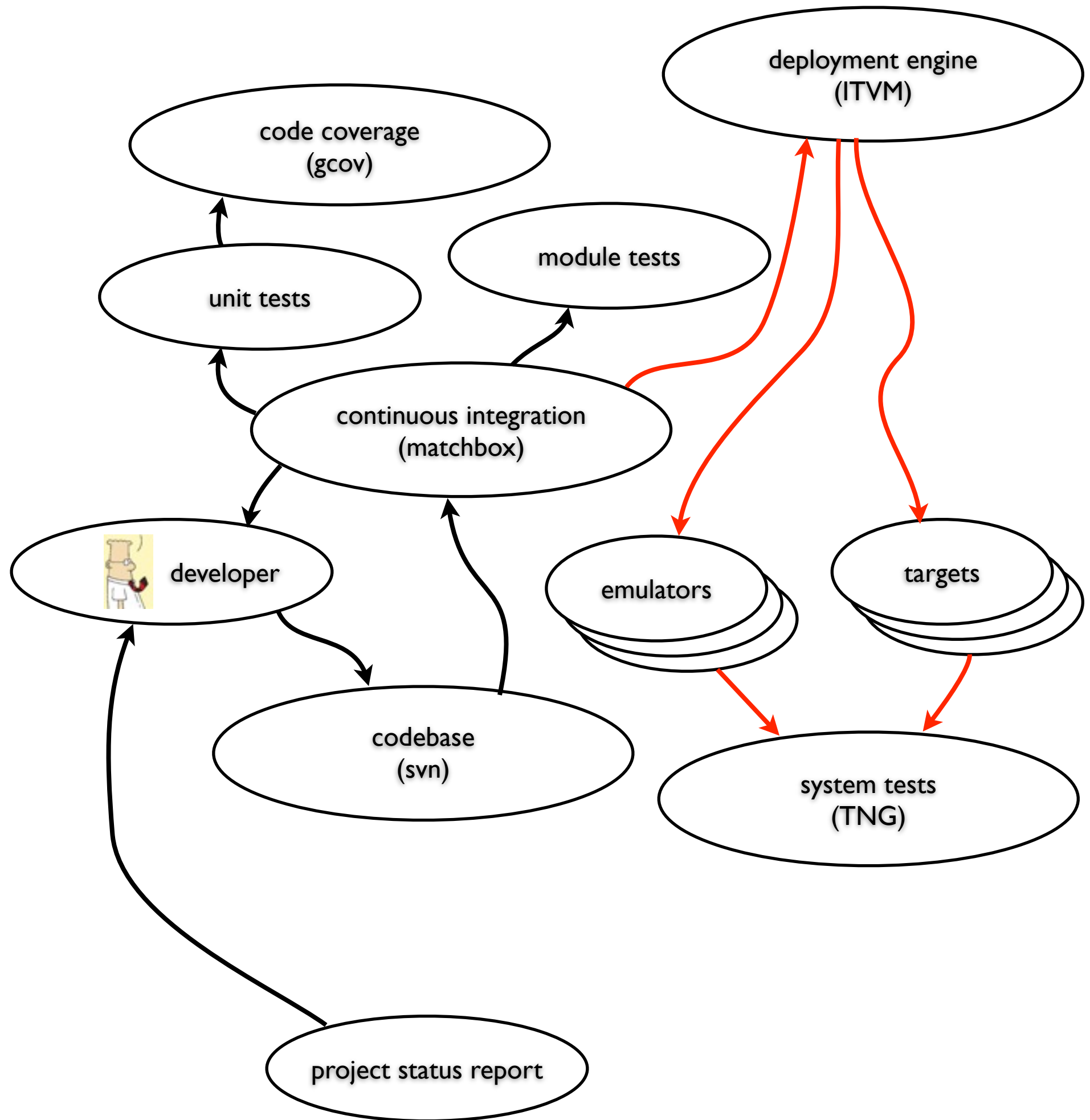


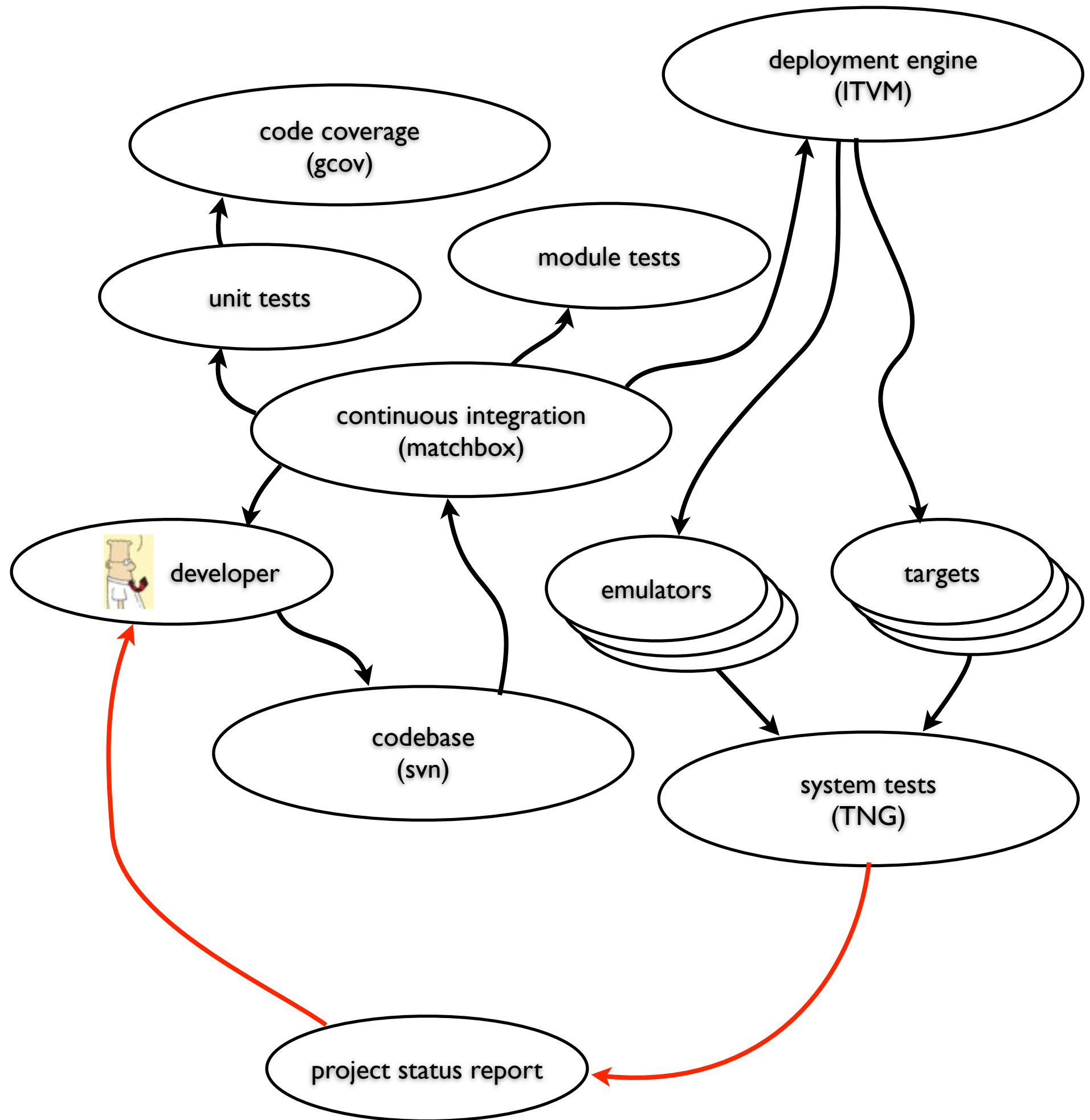


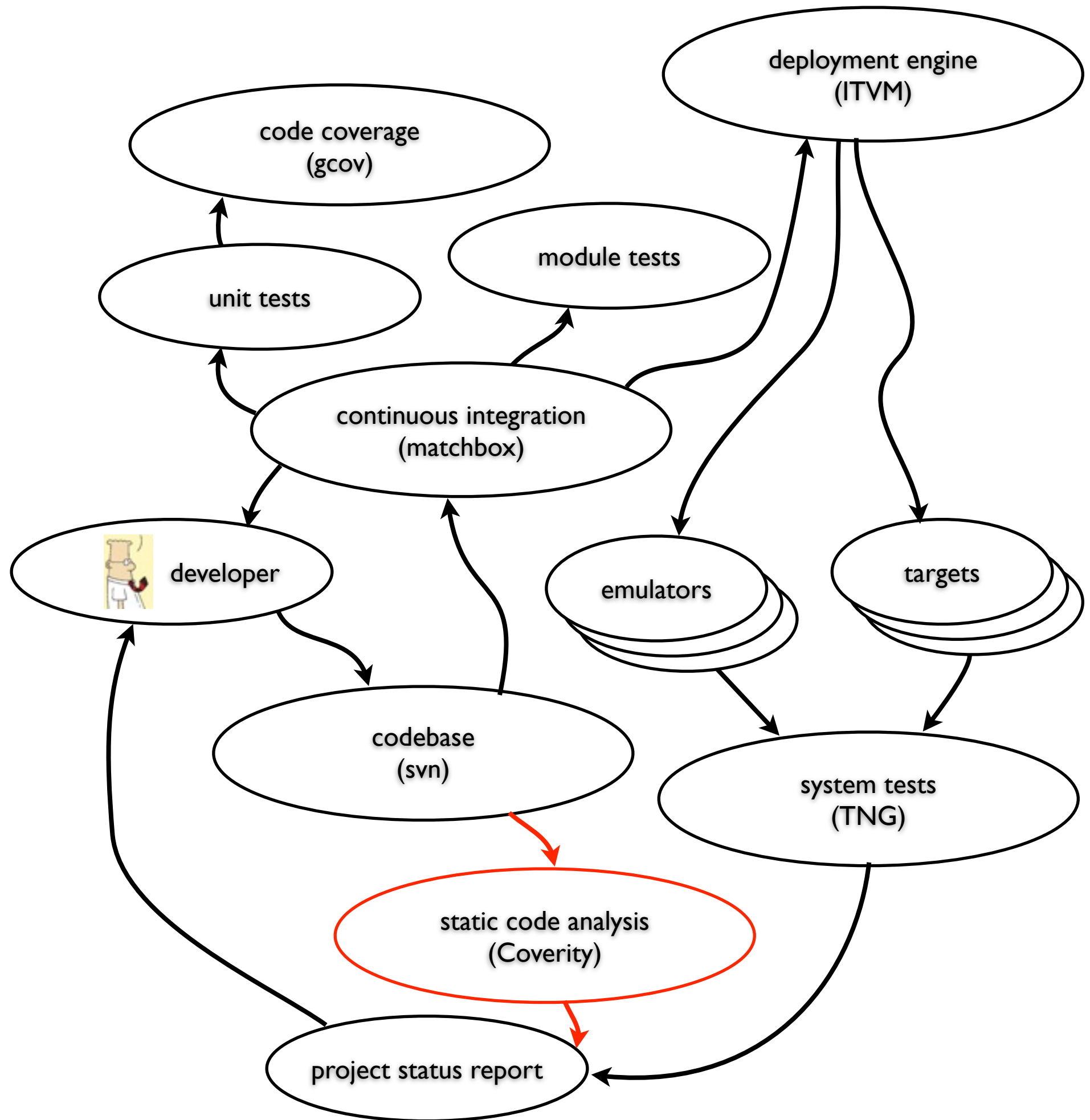


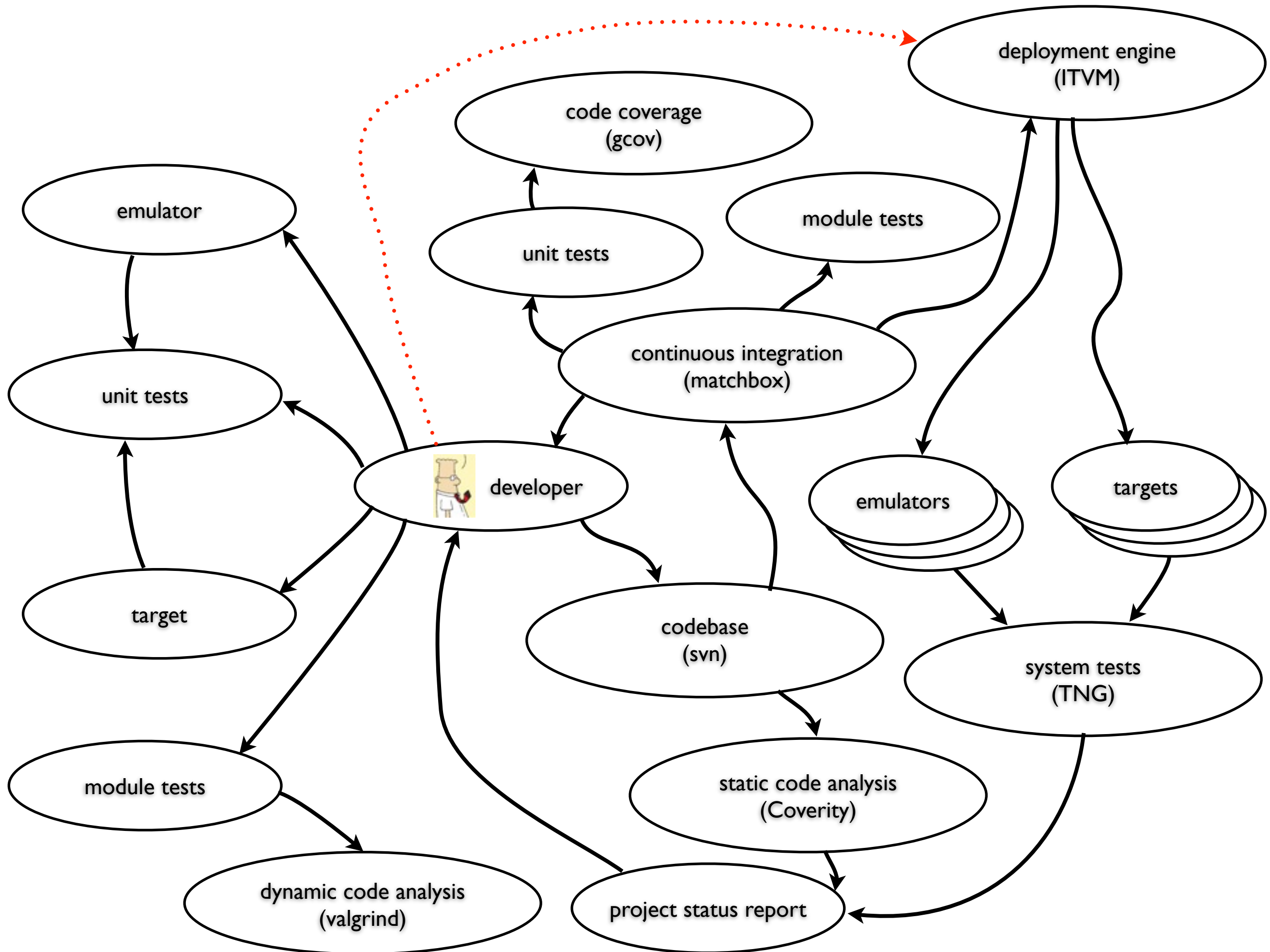


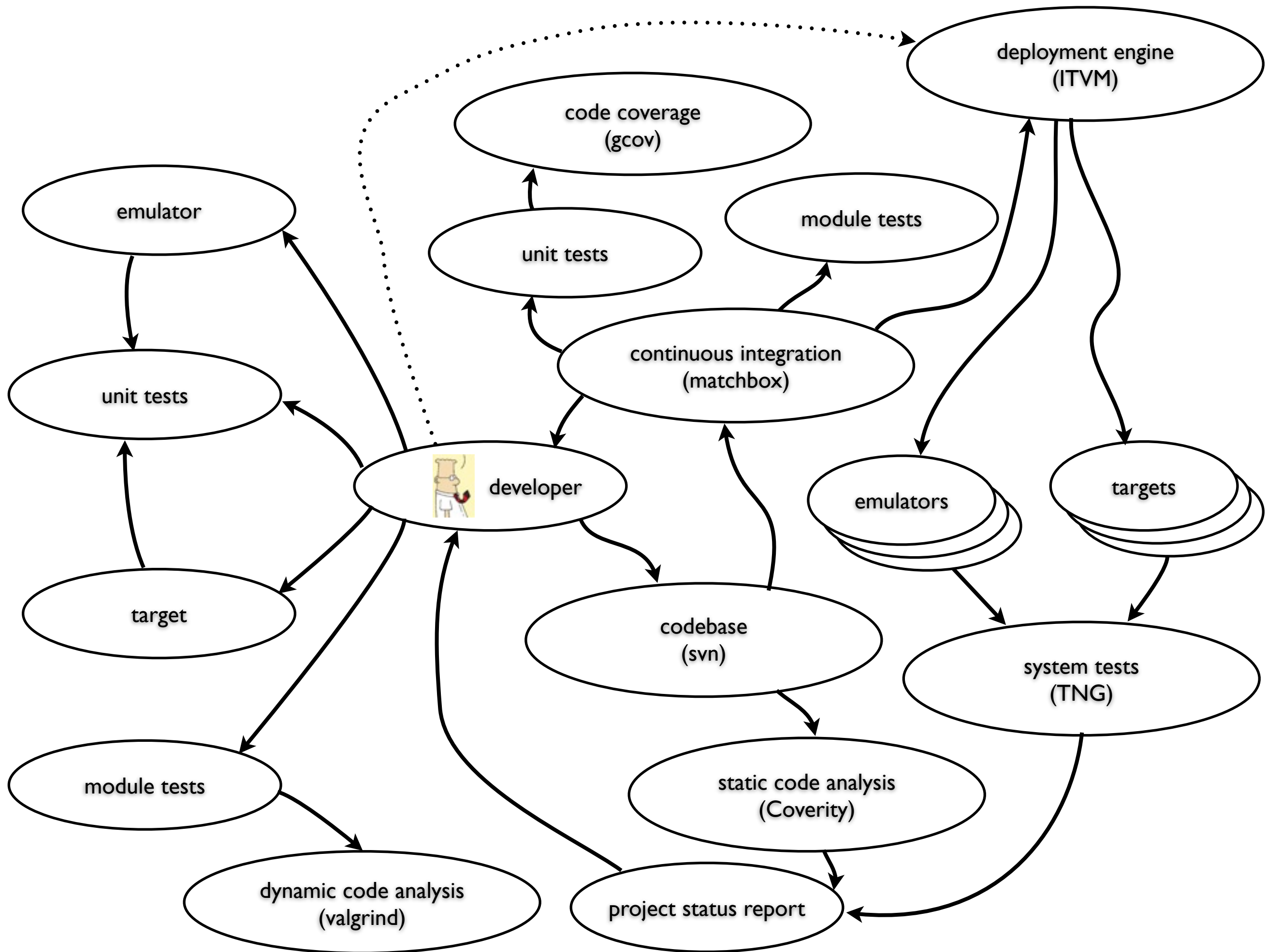












Other aspects of the Saturn project:

- IRC channels
- free choice of development platform
- lot of energy spent on software emulator of actual hardware
- project manager is also configuration manager / build master
- static code analysis (Coverity)
- dynamic code analysis (valgrind)
- build system (genmake2, inhouse python)
- automatic deployment engine (ITVM, inhouse C#)
- automatic system testing (TNG, inhouse python)
- unit test framework (unittest, inhouse C and C++)
- module test framework (inhouse C++)
- code coverage (gcov)
- continuous integration system (matchbox, inhouse python)

Example of visual feedback (HTML pages used by all/most developers)



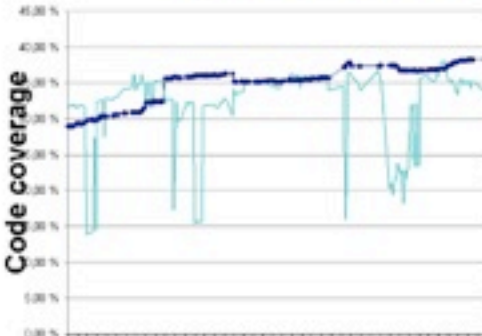
viewvc



irc channel



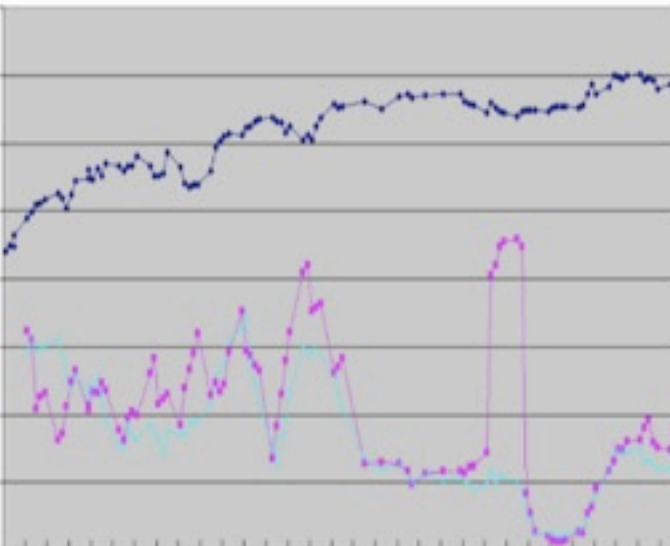
system tests



diff from viewvc



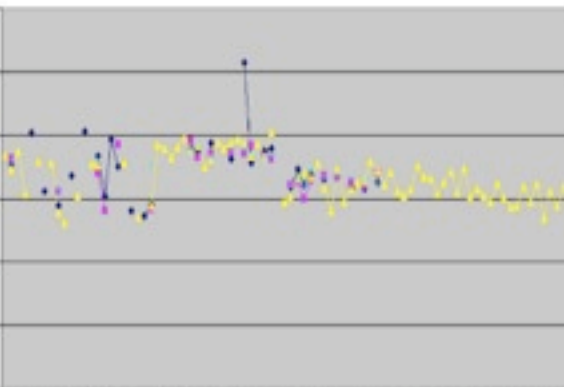
continuous integration



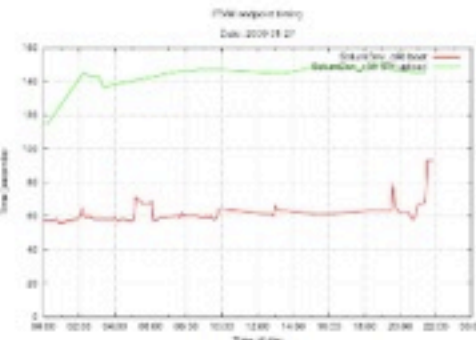
defect trends



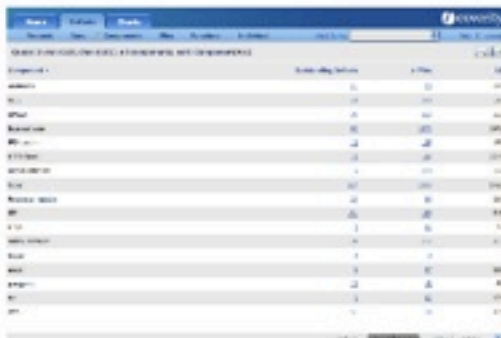
audio delay trend



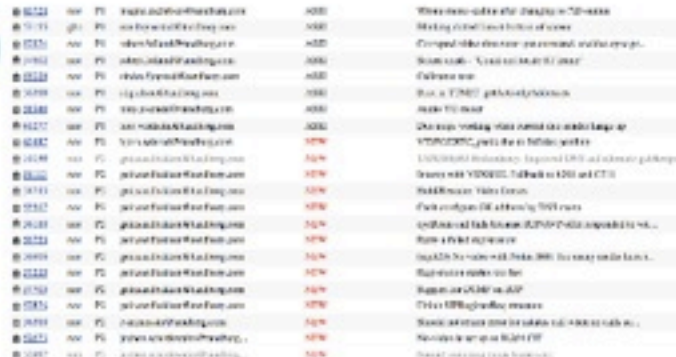
H.264 delay trend



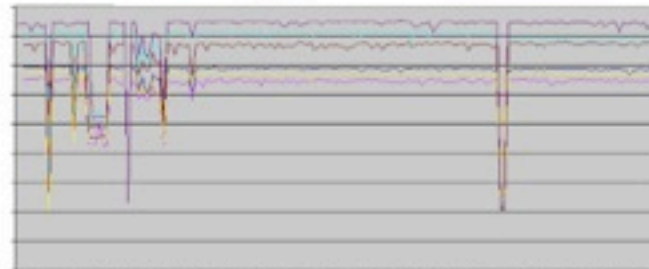
endpoint timing



coverity



bugzilla



PESQ trend

lipsync trend



QA Status

r174136 diff	mho	Bug #61597 - Re-enabled compiler warnings in comcore M /branches/oaktrunk/main/ppcmains/comcore/genmake.def
2009-01-28 17:14:31		
Report Breakage		
r174135	STATUS main	module [tests] rover [boot prodload target] saturn [complete* host-complete* prodtest tilera-complete] skeleton [host tilera] snoopy [complete* host-complete*] testapps [audlocal-host vidcodec*]
r174135 diff	smu	* reverted the last commit D /trunk/main/product/gold/src/test/java/com/tandberg/gold/JavaScriptTest.java
2009-01-28 17:10:39		
Report Breakage		
r174134	STATUS main	module [tests] rover [boot prodload target] saturn [complete* host-complete prodtest tilera-complete] skeleton [host tilera] snoopy [complete* host-complete*] testapps [audlocal-host vidcodec*]
r174134 diff	smu	* Added a junit test to execute the .js tests using Rhino A /trunk/main/product/gold/src/test/java/com/tandberg/gold/JavaScriptTest.java
2009-01-28 17:09:21		
Report Breakage		
r174133	STATUS oaktrunkmain	oak [complete host-complete host-testapps] rkvyf [host]
r174133 diff	mho	Bug #61596 - Re-enabled warnings in lincfg M /branches/oaktrunk/main/ppcmains/lincfg/genmake.def
2009-01-28 17:08:56		
Report Breakage		
r174132	STATUS main	module [tests] rover [boot prodload target] saturn [complete* host-complete* prodtest tilera-complete] skeleton [host tilera] snoopy [complete* host-complete*] testapps [audlocal-host vidcodec*]
r174132 diff	hei	Fix warnings. M /trunk/main/functional/video/codec/nonproduct/rtv/decoder/startcodedec.c
2009-01-28 17:05:37		
Report Breakage		
r174131	STATUS oaktrunkmain	oak [complete host-complete host-testapps] rkvyf [host]
r174131 diff	mho	Bug #61595 - Re-enabled warnings in assent M /branches/oaktrunk/main/ppcmains/assent/genmake.def
2009-01-28 17:02:42		
Report Breakage		
r174130	STATUS bp_famain	fanta [boot host prodtest target*]
r174130 diff	tsk	merged r174118 from bp_camain M /branches/bp_famain/main/ppcmains/h323media/h323mfsm.c M /branches/bp_famain/main/ppcmains/h323stack/h245lofunc.c M /branches/bp_famain/main/ppcmains/h323stack/h245loilcse.c M /branches/bp_famain/main/ppcmains/h323stack/h245loocese.c M /branches/bp_famain/main/ppcmains/h323stack/h245loolcse.c M /branches/bp_famain/main/ppcmains/h323stack/h323ccsfsm.c M /branches/bp_famain/main/ppcmains/h323stack/h323arsfsm.c M /branches/bp_famain/main/ppcmains/h323stack/q931message.c M /branches/bp_famain/main/ppcmains/h323stack/rasfunc.c M /branches/bp_famain/main/ppcmains/h323stack/rassfsm.c M /branches/bp_famain/main/ppcmains/miscutil/api/h323mediautil.h M /branches/bp_famain/main/ppcmains/miscutil/api/mediacaps.h M /branches/bp_famain/main/ppcmains/miscutil/h323mediautil.c M /branches/bp_famain/main/ppcmains/miscutil/mediacaps.c
2009-01-28 16:57:05		
Report Breakage		

About Software Development

We believe that **software development** is a continuous learning process and a cooperative game of communication between professionals. Software development is about repeating cycles of preparing, changing, observing, reflecting, and learning.

We believe that software development is a **continuous learning** process and a cooperative game of communication between professionals. Software development is about repeating cycles of preparing, changing, observing, reflecting, and learning.

We believe that software development is a continuous learning process and a **cooperative game of communication** between professionals. Software development is about repeating cycles of preparing, changing, observing, reflecting, and learning.

We believe that software development is a continuous learning process and a cooperative game of communication between **professionals**. Software development is about repeating cycles of preparing, changing, observing, reflecting, and learning.

We believe that software development is a continuous learning process and a cooperative game of communication between professionals. Software development is about repeating cycles of preparing, changing, observing, reflecting, and learning.

We believe that software development is a continuous learning process and a cooperative game of communication between professionals. Software development is about repeating cycles of preparing, changing, observing, reflecting, and learning.



Few software projects are like running on a paved road where you can see the ...



... goal in the end of the road.

Most projects are more like...



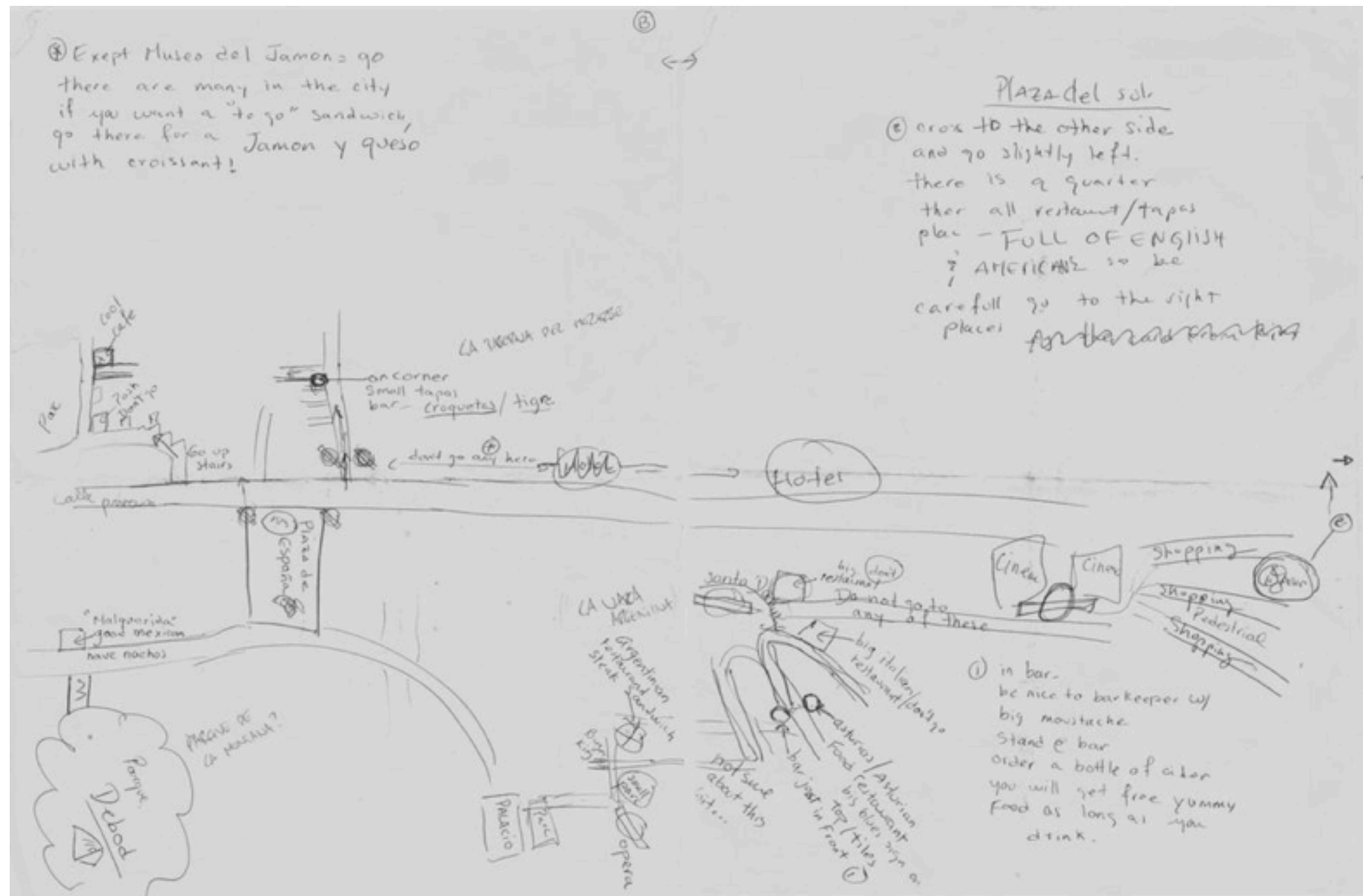
extreme orienteering



with a group of people



in the dark



with only a sketchy map as guidance



Problems in software development usually multiply and gets worse by exerting more control...



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

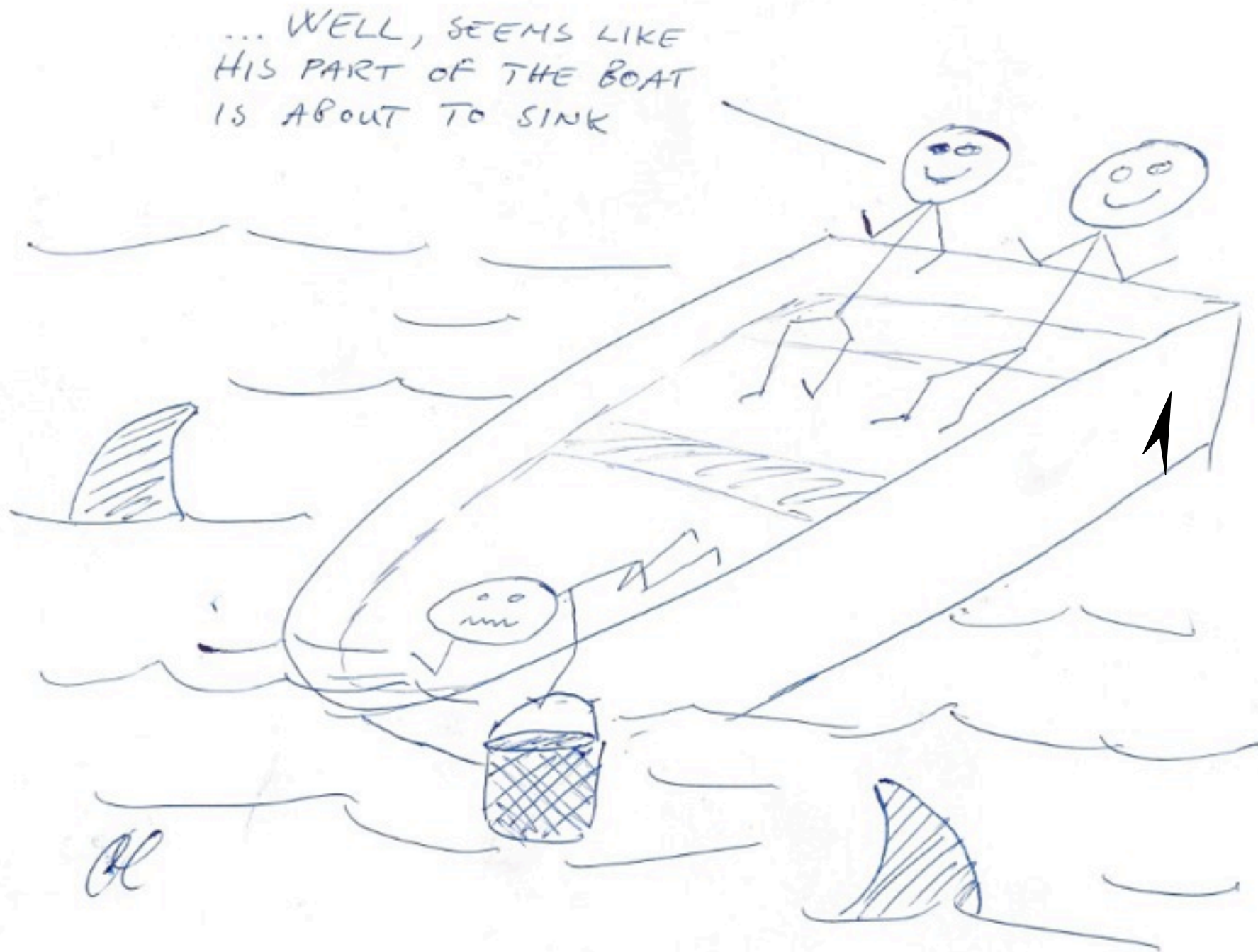
(Princess Leia)

!

Appendix

About Global Development

Make sure that everybody is working towards a common goal.



Scaling into global development

- rule 0: avoid multisite development, if you can...
- visual communication is essential
- time difference is more challenging than geographical separation
- use continuous integration and automatic testing
- working across trust boundaries is painful
- balance the sites
- cultural differences is more challenging than time difference
- prefer local decisions
- accept duplication of work, beware the share
- move people around, use ambassadors
- beware of the Cover My Ass game
- avoid contracted interfaces, use mediators
- use tools for microcommunication
- focus on system architecture
- knowledge management
- corporate culture

Some random quotes

"What we don't do is treat our employees like they're all, you know, criminals,"

(Jenn Mann, SAS Institute)

"Controlling an organization by monitoring costs is like driving a car looking out the rear window."

(John Seddon)

Never tell people how to do things. Tell them what to do and they will surprise you with their ingenuity.

(General George Patton Jr)

Those who built the good-to-great companies made as much use of "stop doing" lists as "to do" lists. They displayed a remarkable discipline to unplug all sorts of extraneous junk. (Collins, 2001)

Discipline by itself will not produce great results. We find plenty of organizations in history that had tremendous discipline and that marched right into disaster, with precision and in nicely formed lines. (Collins, 2001)