

Verdens beste utviklingsmiljø

v/ Olve Maudal, TANDBERG
olve.maudal@tandberg.com

TANDBERG har aldri brydd seg mye om dokumentasjon, prosedyrer, metodikker og risikoreduksjon. Derimot har vi stor tro på at vår kultur og prinsipper gjør oss i stand til å utvikle og levere det ene spektakulære produktet etter det andre. TANDBERG har i 10-15 år jobbet etter prinsipper som i dag gjerne omtales som Agile og Lean. I denne presentasjonen vil jeg snakke om produktutvikling i TANDBERG og dele erfaringer og ideer fra et konkret utviklingsprosjekt. Fokus vil være på programvareutvikling.

En presentasjon for System
Tirsdag 12. januar 2010

About Olve

geek - very proud of being a computer programmer.

Have been programming nearly every day since I bought my first computer - the VIC 20.



Studying Software Engineering (Manchester 1992-1995), Artificial Intelligence (Edinburgh 1995-1996) and some postgrad studies in Knowledge Discovery (NTNU 1996). After graduating I started developing systems for finding oil (Schlumberger 1996-2000), then I developed systems for moving money (BBS 2000-2004), and now I am developing systems for effective communication between people (Tandberg since 2004).

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 1989. Since then TANDBERG has grown from a small startup based in Norway into an international company with ~1600 employees and a revenue of 800 MUSD in 2008. ~450 engineers work with product with research and development.

Dual headquarters in New York and Oslo. R&D centres in Oslo (NO), Langley (UK), Ruscombe (UK), Bangalore (IN), Hamilton (NZ) and Sydney (AU).

www.tandberg.com

Breaking news (Dec 4, 2009): After a successful \$3.4bn bid Cisco now controls more than 90% of shares in TANDBERG. The transaction is expected to close some time during 2010.



TANDBERG

www.tandberg.com

~~Tandberg Data~~

~~Tandberg Storage~~

~~TANDBERG Television~~

~~TANDBERG Display~~

~~Tandberg Eiendom~~

~~Robert Tandberg Møbler~~



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

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

We develop and sell...



Meeting room systems



Telepresence systems



Personal systems



PC based solutions



Networking products



And a lot of other stuff



~1600 employees worldwide

~450 R&D engineers

~250 write code every day

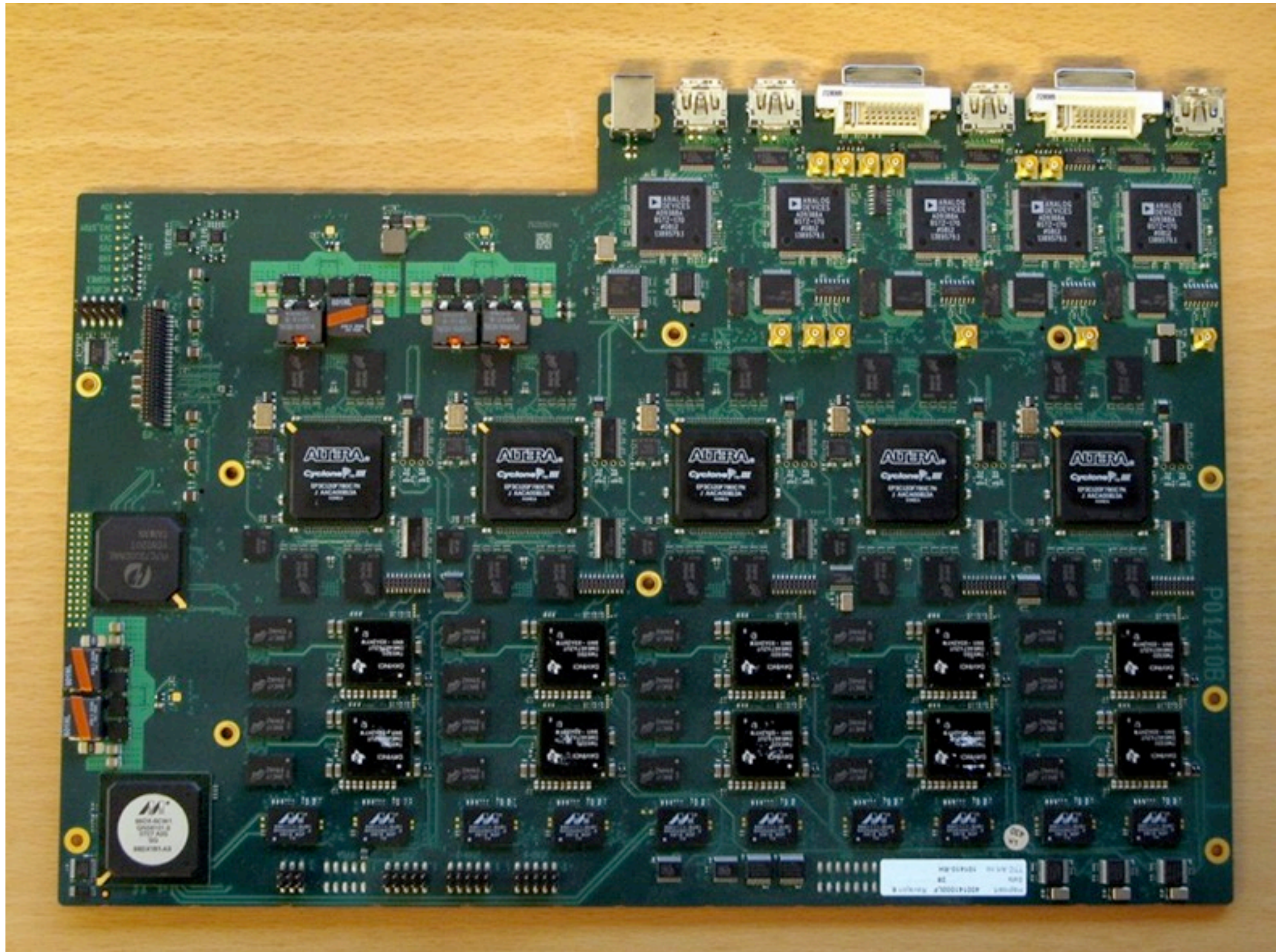
C, C++, Python, Java, C#, VHDL, Ruby

most of us work with software development

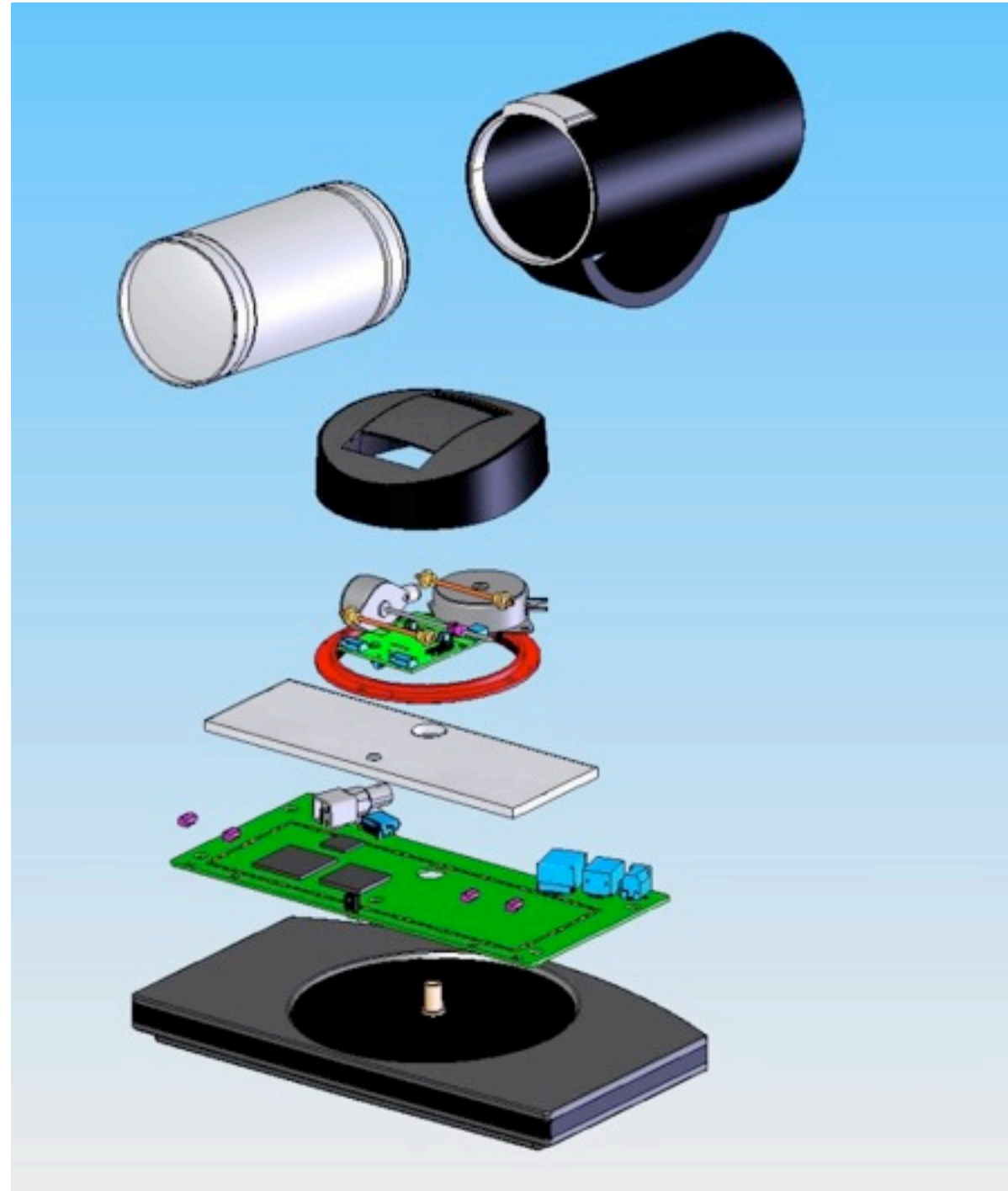


but we also do...

Electronics / Hardware



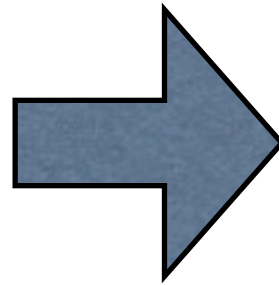
Mechanics



Industrial Design



1992



2007

Looking into



the future

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. As we will discuss later; every project, product and team is unique, thus it does not make sense to create a particular procedure to follow.

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>



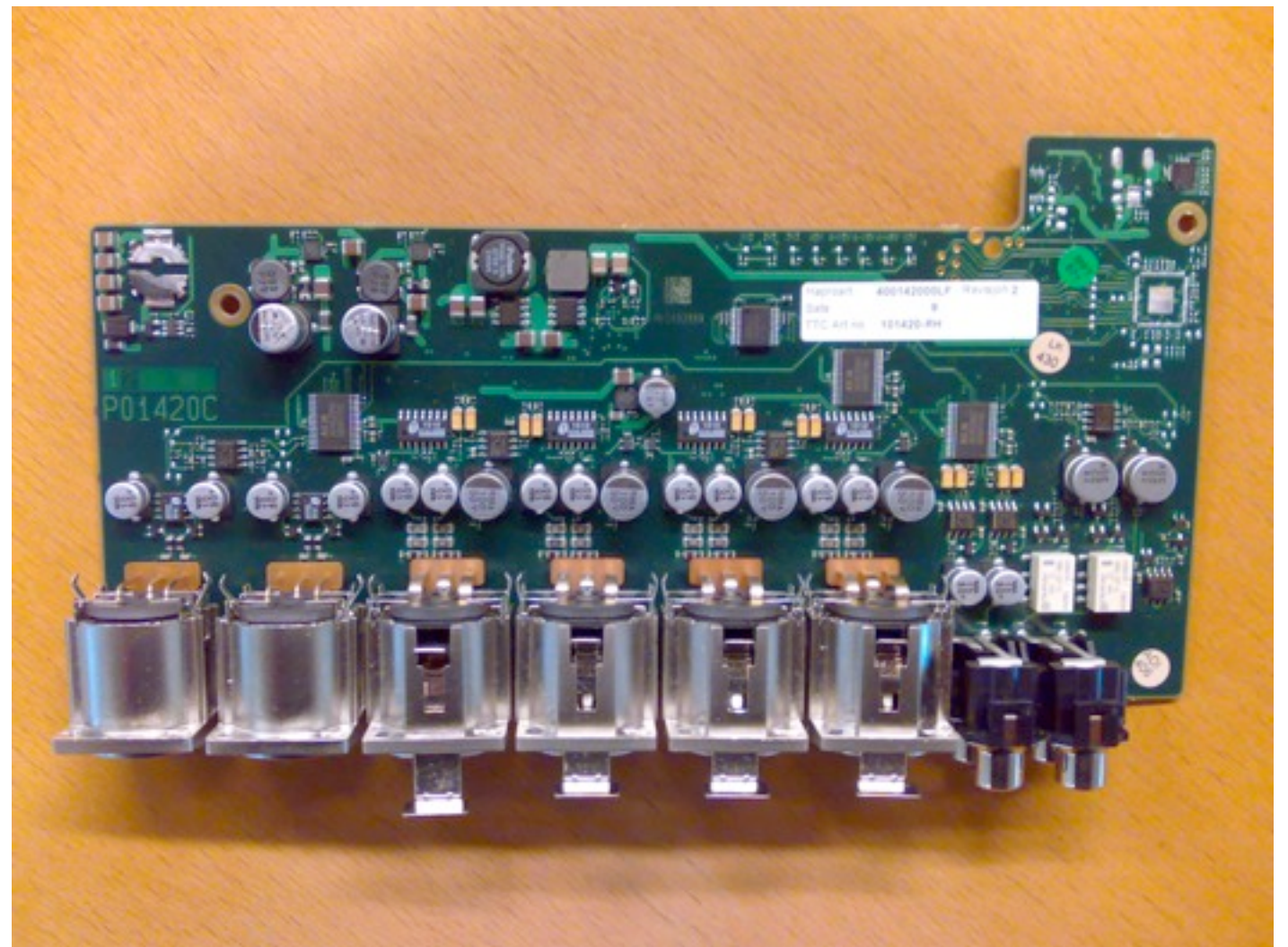
C90 Features:

- 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



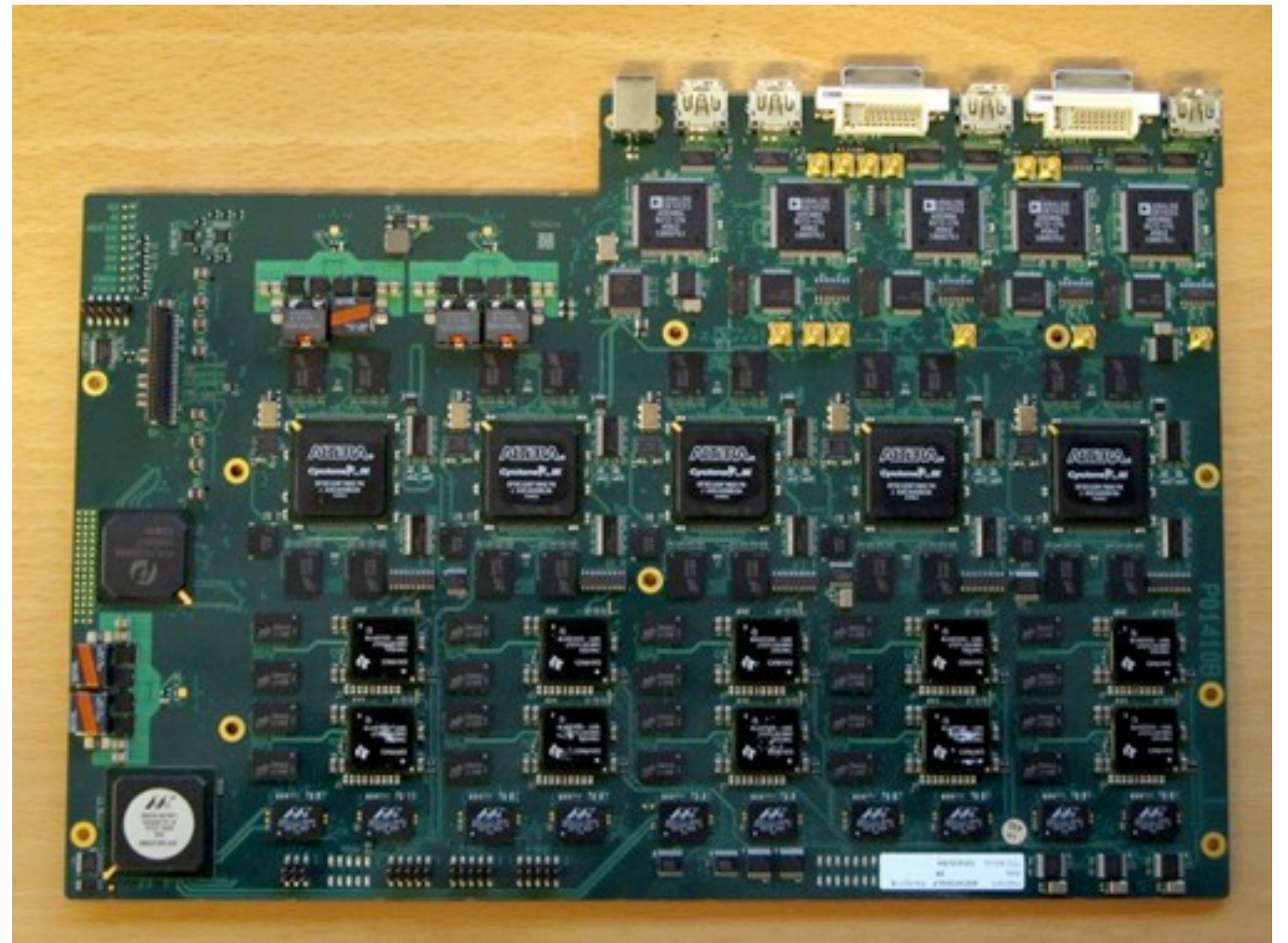
C90 AUDIO EXTENSION BOARD

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



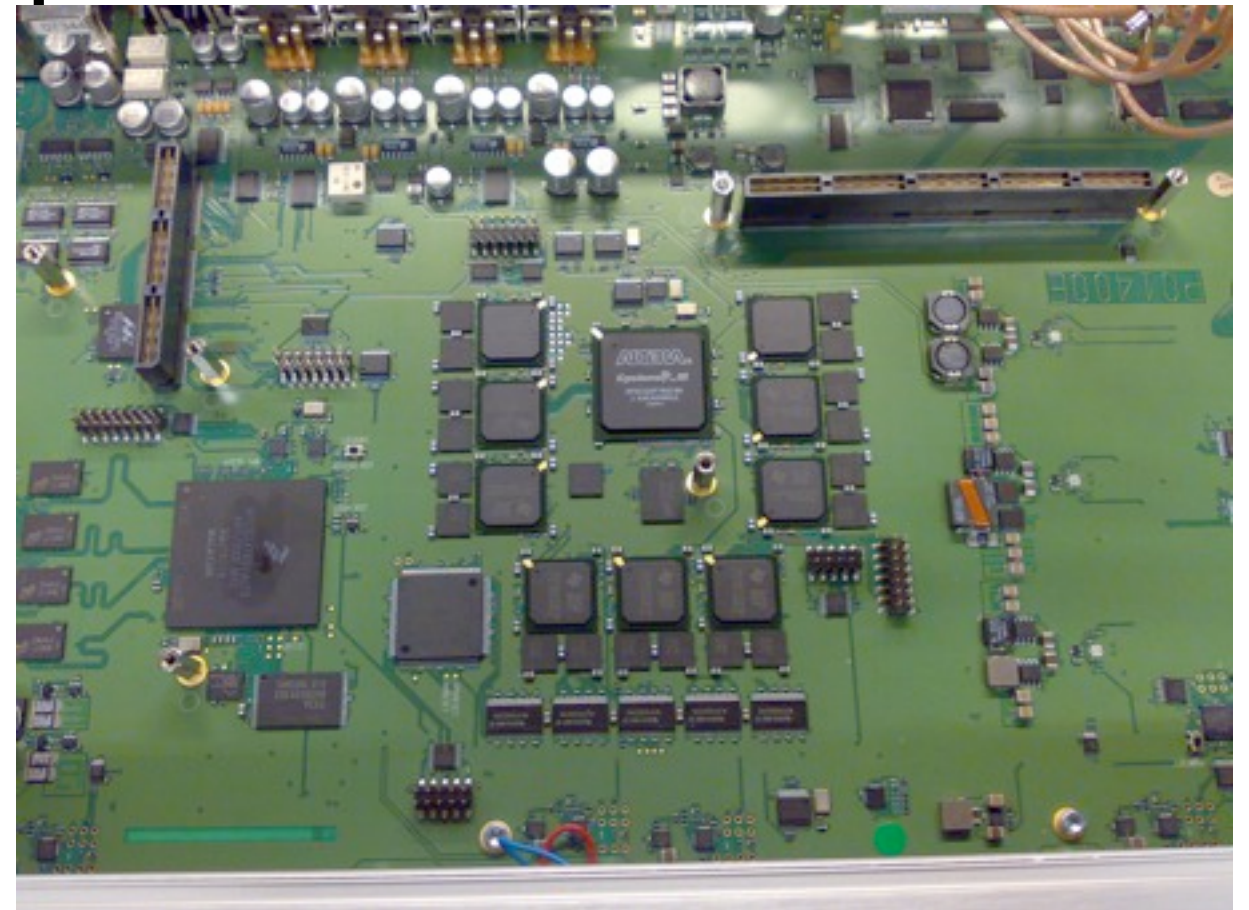
C90 VIDEO BOARD

- 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

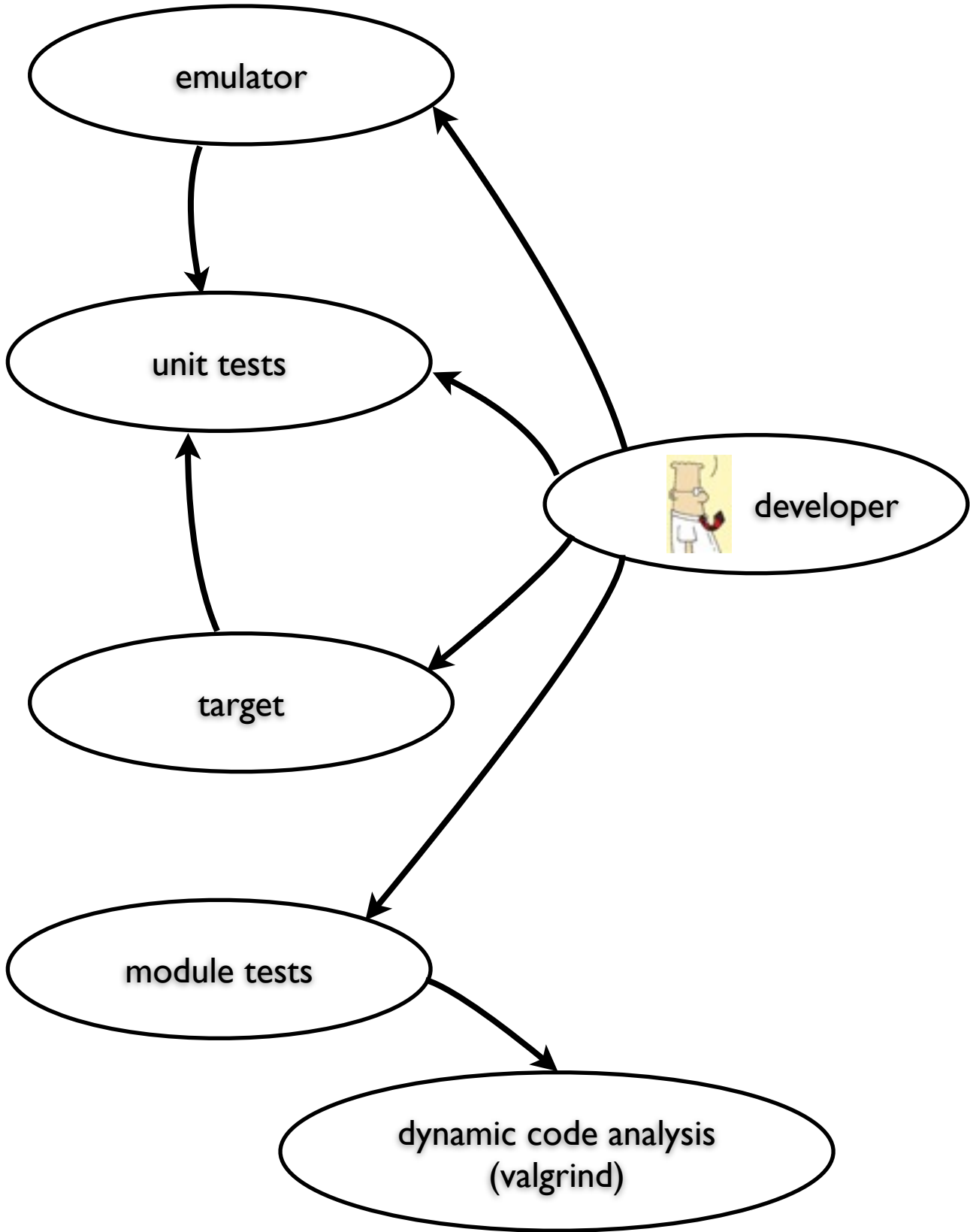
Development Practices in the Saturn project

- Continuous planning
- Always attack high risks first
- Heavy focus on effective feedback mechanisms
- Visualization of actual status throughout project
- Parallel development
- Iterations and time-boxing
- Daily scrum of scrums
- Weekly rendezvous meetings
- Early and many prototypes

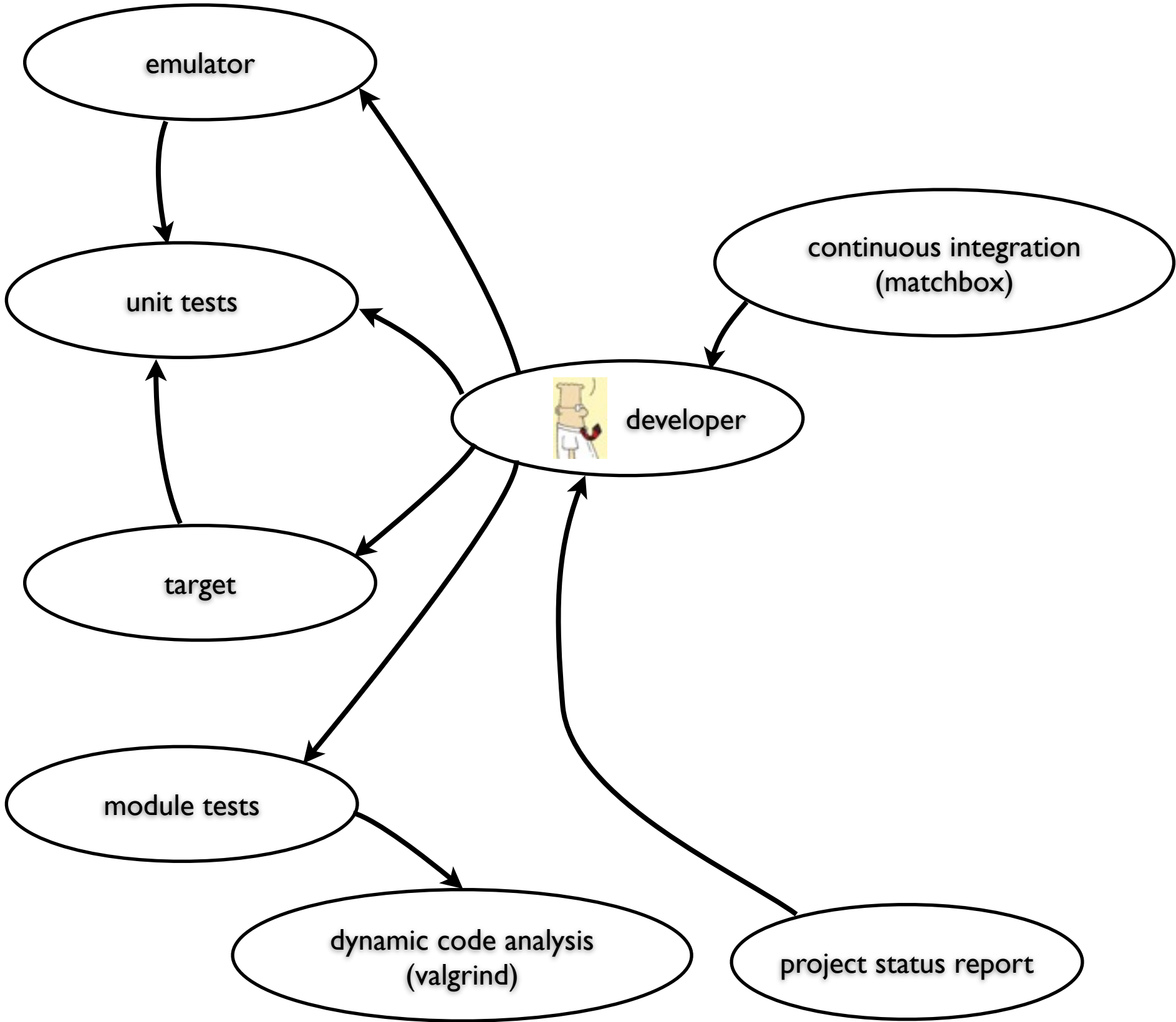
Typical workflow in the Saturn project -
seen from a developers' point of view



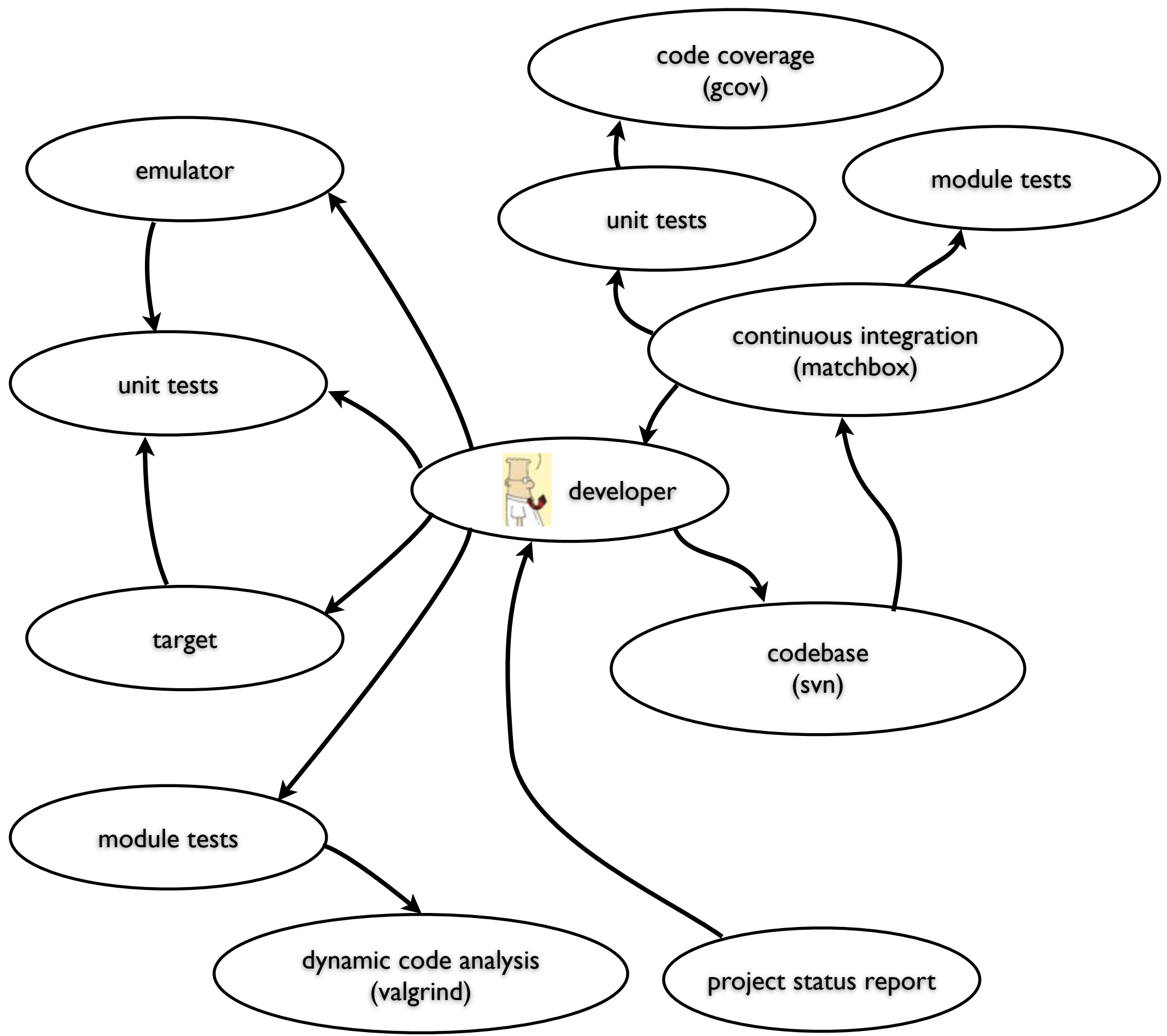
Typical workflow in the Saturn project -
seen from a developers' point of view



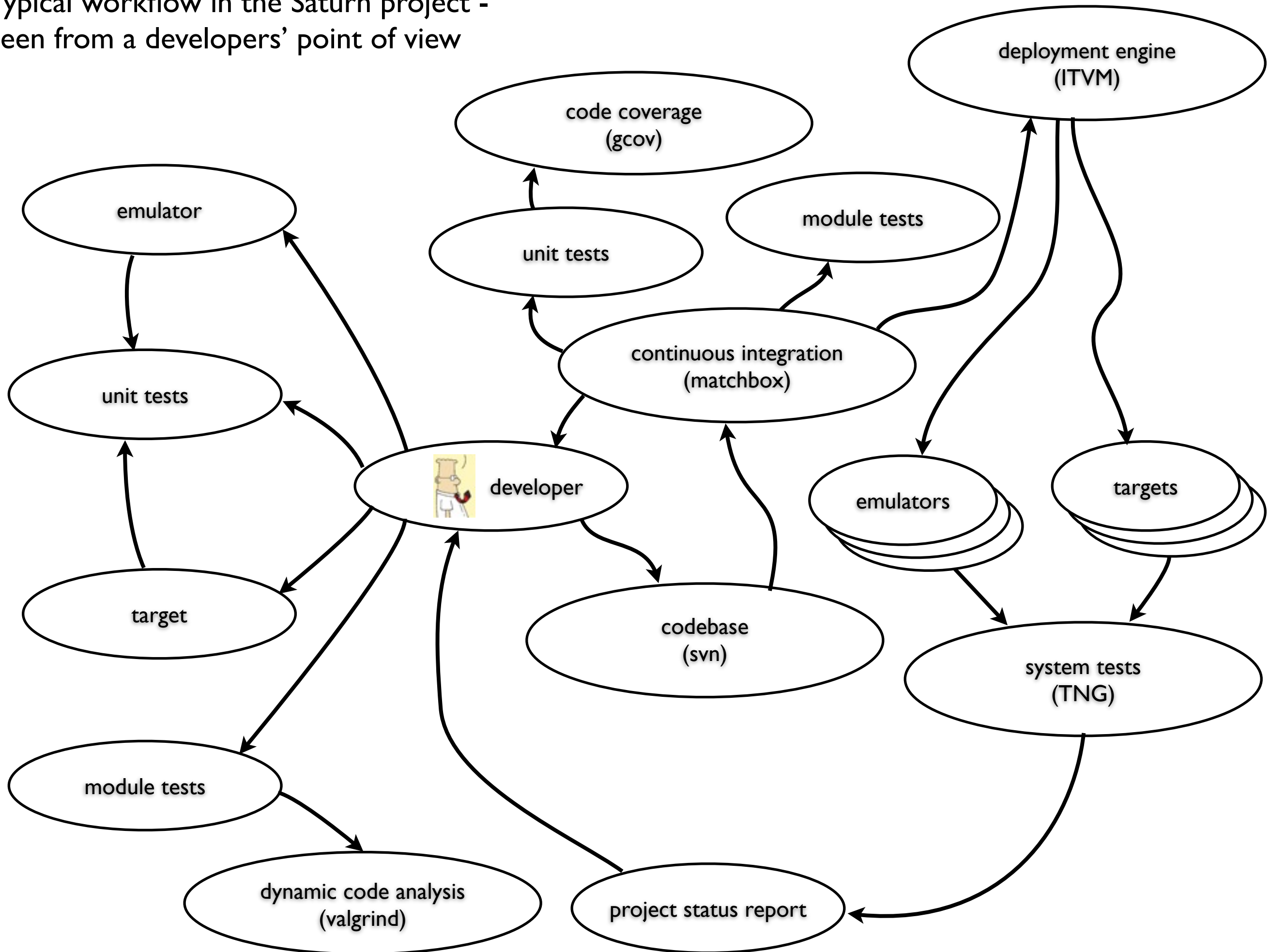
Typical workflow in the Saturn project -
seen from a developers' point of view



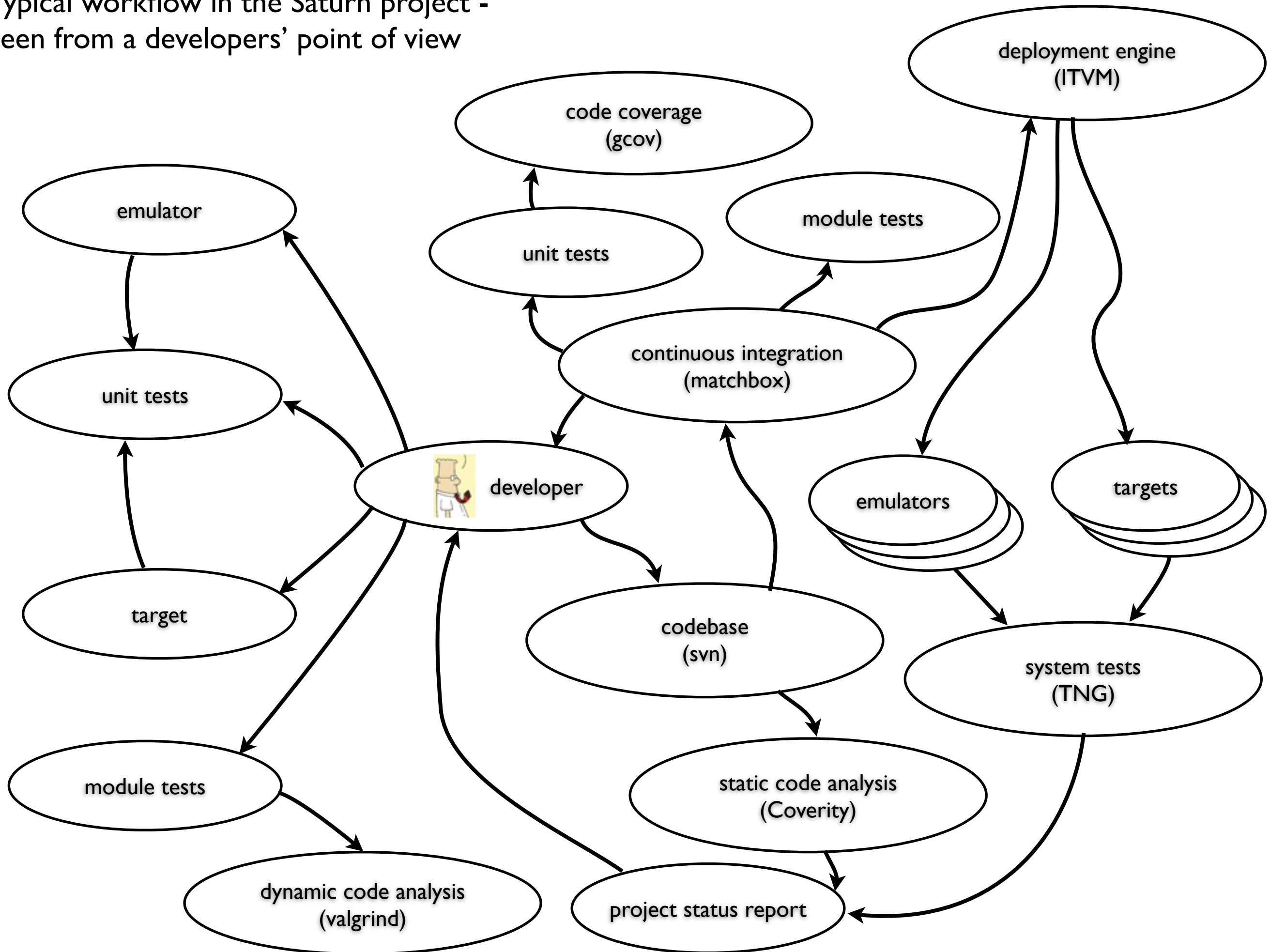
Typical workflow in the Saturn project -
seen from a developers' point of view



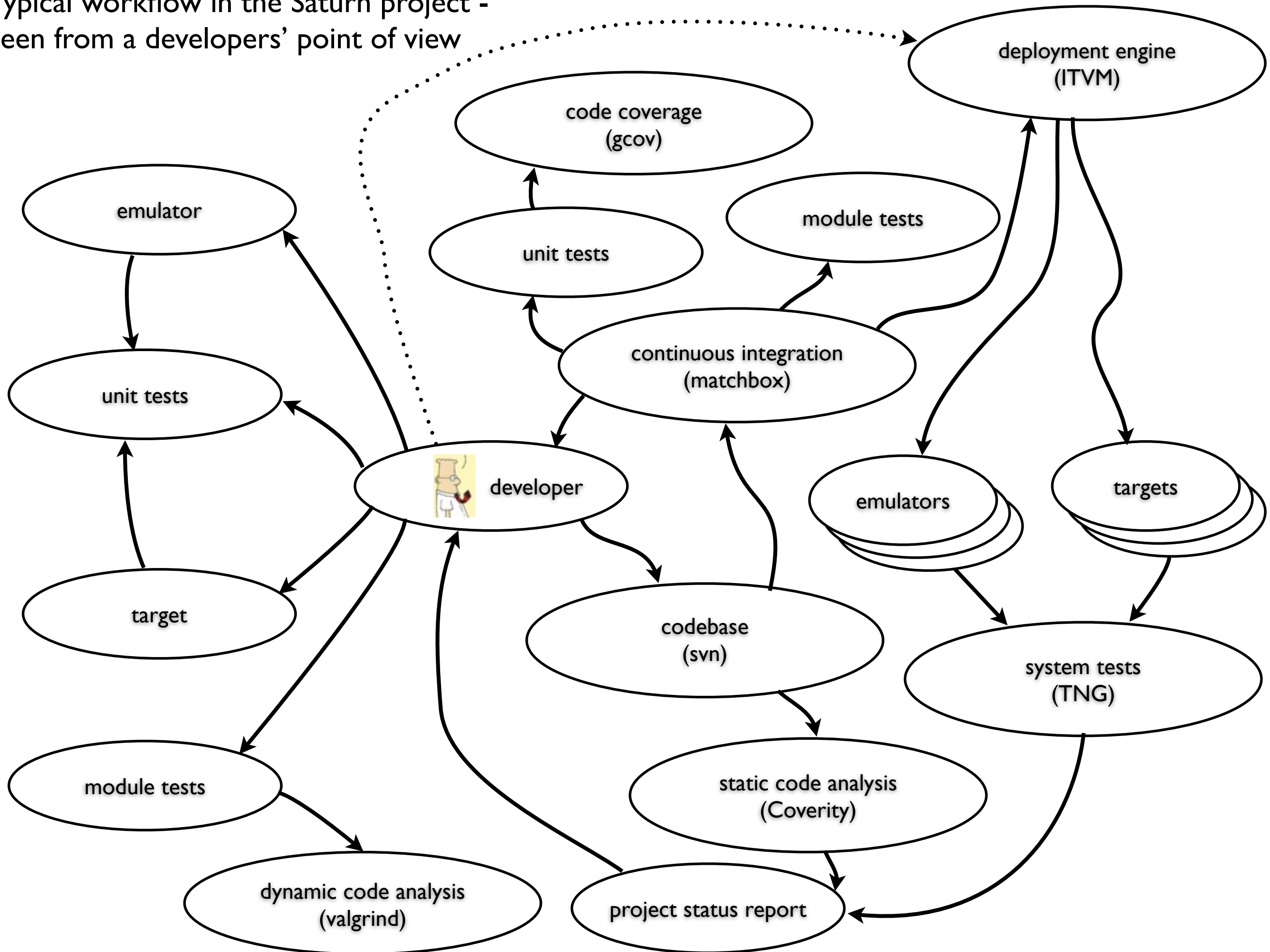
Typical workflow in the Saturn project -
seen from a developers' point of view



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



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



Other aspects of the Saturn project:

- IRC channels
- free choice of development platform
- software repository (svn)
- warning free compilation!
- proper training (C++, C, Testing, Professionalism, ...)
- lot of energy spent on software emulator of actual hardware
- project leader is also the configuration manager / build master
- Teams: GUI, App, Protocol, Video, Audio, FPGA, Platform, QA, Support
- 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)



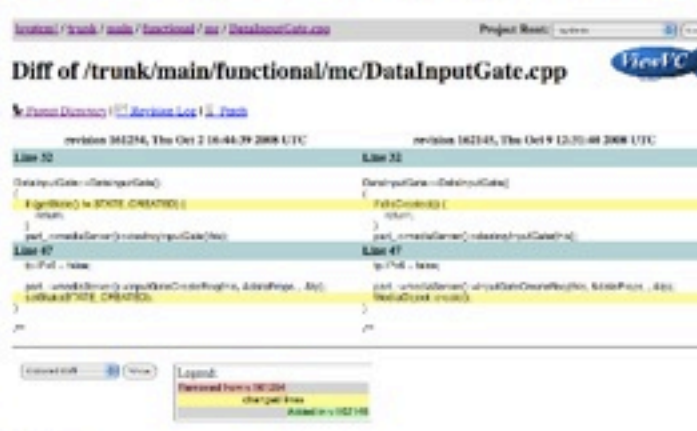
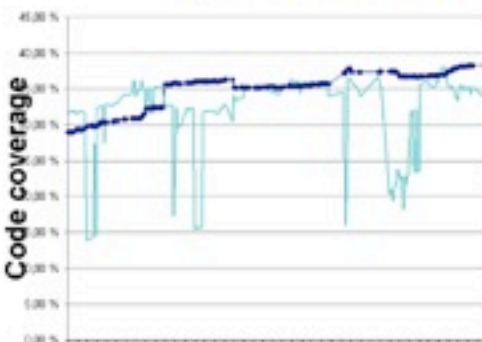
viewvc



irc channel



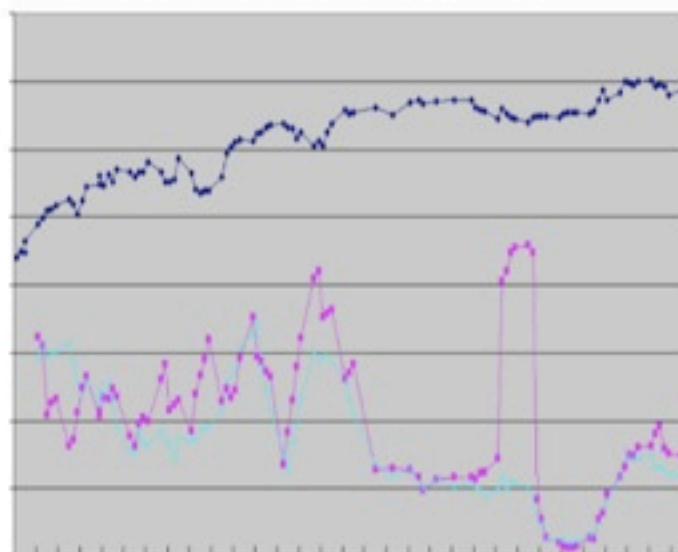
system tests



diff from viewvc



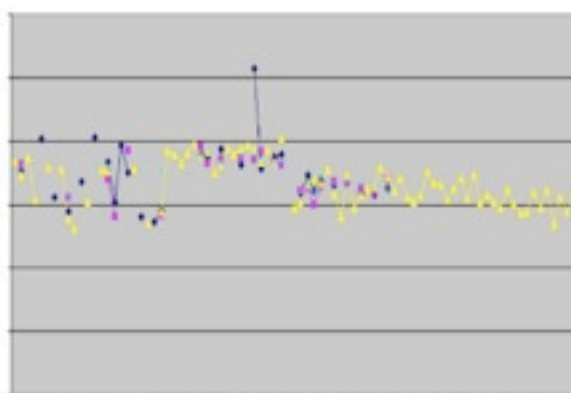
continuous integration



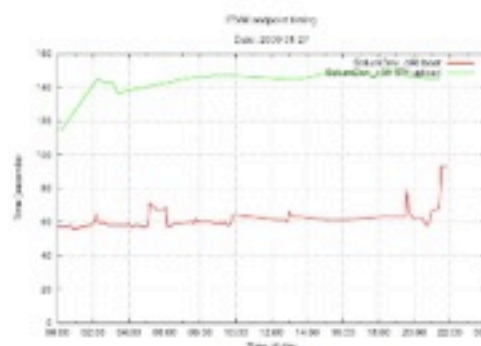
defect trends



audio delay trend



H.264 delay trend



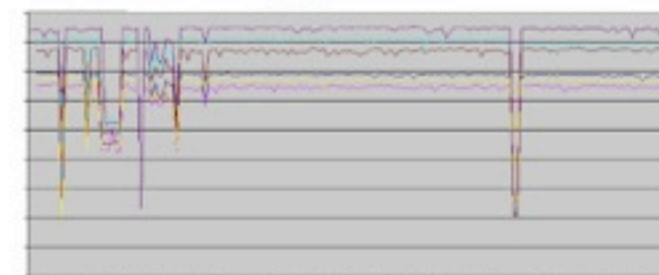
endpoint timing



coverity



bugzilla



PESQ trend

lipsync trend



QA Status

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.

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.

So... for this project...

When we saw an opportunity to improve feedback mechanisms within the project we did. When we found things that reduced communication within the project we tried to get rid of it.

Especially in crunch mode, we focused on making sure that the feedback mechanisms was working. Automated feedback is about giving developers confidence when making changes - when confidence is lost, everything is lost...

Product Development in TANDBERG

- Little documentation gives effective communication
- No corporate standards or procedures
- Treat engineers as professionals, not as resources
- Spend very little energy on things that are not essential
- "Plans are nothing, planning is everything"
- Do not write hours, do not measure project cost
- Delay decisions as much as possible
- To fail is OK, therefore we deliver spectacular stuff
- Doers are very much respected in Tandberg
- We hire and keep exceptional people
- Communication is a key skill for all our engineers
- We are fast and “sloppy”
- We release early and we release often
- Slack is embedded, and “skunk work” projects appreciated
- The company builds on trust
- Fun and Profit

Product Development in TANDBERG

- Little documentation gives effective communication
- No corporate standards or procedures
- Treat engineers as professionals, not as resources
- Spend very little energy on things that are not essential
- "Plans are nothing, planning is everything"
- Do not write hours, do not measure project cost
- Delay decisions as much as possible
- To fail is OK, therefore we deliver spectacular stuff
- Doers are very much respected in Tandberg
- We hire and keep exceptional people
- Communication is a key skill for all our engineers
- We are fast and “sloppy”
- We release early and we release often
- Slack is embedded, and “skunk work” projects appreciated
- The company builds on trust
- Fun and Profit

We follow principles, not processes!

Appendix

It takes a lot to stay ahead in the game...

Inhouse training

C Foundation course

C++ Foundation course

C++ Advanced course

UML Course

Pattern-Based Software Development Course

Studygroups in OOAD

Studygroups in Java

Studygroups in C++

Agile Planning, Estimation and Retrospectives

Certified ScrumMaster

Agile Modeling Design Workshop

Software Architecture

Pattern Language of Tandberg

Python course

Scala course

... and much more

Hackers' Corner - Internal seminars



Inhouse conferences

TechZone Lillehammer 2007

~ 190 engineers, 32 talks, 4 tracks

TechZone Lysaker 2007

~ 260 engineers, 40 talks, 5 tracks

TechZone Barcelona 2008

~ 380 engineers, 43 talks, 5 tracks

TechZone Storefjell 2010

~ 500 engineers, 45 talks, 5 tracks, TechFair

Engage industry experts...

Dana Bredemeyer - Software Architecture



Kevlin Henney

C++, C, UML, Pattern-Based Software Development, Pattern Language of Tandberg



Michael Feathers

Pair-Programming, Testing, Professionalism



Jon Jagger - C, C++, UML, Patterns, Agile, TDD



Jutta Eckstein
Project Management, Agile Planning, Retrospectives



Robert C. Martin “Uncle Bob”
TDD, Pair Programming, Professionalism, Agile Design Course (SOLID)



Tom and Mary Poppendieck

Project Management, Lean Software Development



Craig Larman

Agile Modeling Design Workshop



David Beazley - Python training



Geir Amsjø - Certified ScrumMaster Course



Bruce Perens - Free and Open Source issues



Jonas Boner - Scala training

Example - Spring clustering

Terracotta config

```
<spring>
  <application name="tc-jmx">
    <application-contexts>
      <application-context>
        <paths>
          <path>*/applicationContext.xml</path>
        </paths>
        <beans>
          <bean name="/counter">
            <bean name="/history">
            </bean>
          </bean>
        </beans>
      </application-context>
    </application-contexts>
  </application>
</spring>
```

- Terracotta beans (Singleton + Session)
- Context events, JMX State and

Spring config

```
<bean id="localCounter"
      class="demo.jmx.Counter"/>
<bean id="clusteredCounter"
      class="demo.jmx.Counter"/>
<bean id="localHistory"
      class="demo.jmx.HistoryQueue"/>
<bean id="clusteredHistory"
      class="demo.jmx.HistoryQueue"/>
```

ing Through Network-Attached Memory

Bjarne Stroustrup - About past and future of C++



RMS



“I have been working with software development groups all around the world, and you are way ahead of most.”

(a consultant visiting our R&D department)



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

Principles behind the Agile Manifesto

We follow these principles:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

The 7 Lean Software Development Principles

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

(Poppendieck)

TANDBERG

SPEED AND PRECISION

Simplify – focus – act • Approximately right rather than accurately wrong • Think • Do it right the first time

INTEGRITY AND ENTHUSIASM

Sense of humour • Honesty • High ethical standards • Excitement • Trustworthiness • Loyalty

EXCEED EXPECTATIONS

Personal initiative • Fighting spirit • Go the last mile

FUN AND PROFIT

Maximize long term shareholder value • Pass på penga • One for all, all for one • Energy

TANDBERG FIRST

First in user benefits • Innovative • "Kreativ galskap" • Understanding customer needs

SPEED AND PRECISION

Simplify - focus - act * Approximately right rather than accurately wrong * Think * Do it right the first time

INTEGRITY AND ENTHUSIASM

Sense of humour * Honesty * High ethical standards * Excitement * Trustworthiness * Loyalty

EXCEED EXPECTATIONS

Personal initiative * Fighting spirit * Go the last mile

FUN AND PROFIT

Maximize long term shareholder value * Pass på penga * One for all, all for one * Energy

TANDBERG FIRST

First in user benefits * Innovative * "Kreativ galskap" * Understanding customer needs

***”We have to grow without sacrificing
our culture and our values”***

***Jan Chr. Opsahl
Chairman***