Software Development at TANDBERG

Case study: TANDBERG Codec C90

March 2009

Olve Maudal, olve.maudal@tandberg.com

(last updated: March 31 2009)

About: Olve Maudal

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) and Artificial Intelligence (Edinburgh 1995-1996). 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 2004-now).

Active member, participant and/or contributor on: JavaPils, Smidig, JavaZone, XP Meetup, Lean Meetup, Rubberducks, Oslo C++ Users Group, ACCU ... and a lot of other things

Blogs regularly on http://olvemaudal.wordpress.com/ and http://twitter.com/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 it's 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 approx 1500 employees and a revenue of 800 MUSD in 2008.
Dual headquarters in New York and Oslo. Development centres in Oslo, Langley, Ruscombe, Bangalore, New Zealand and Sydney.

www.tandberg.com

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

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

TANDBERG

www.tandberg.com

TANDBERG Television

Tandberg Data

Tandberg Storage

TANDBERG Display

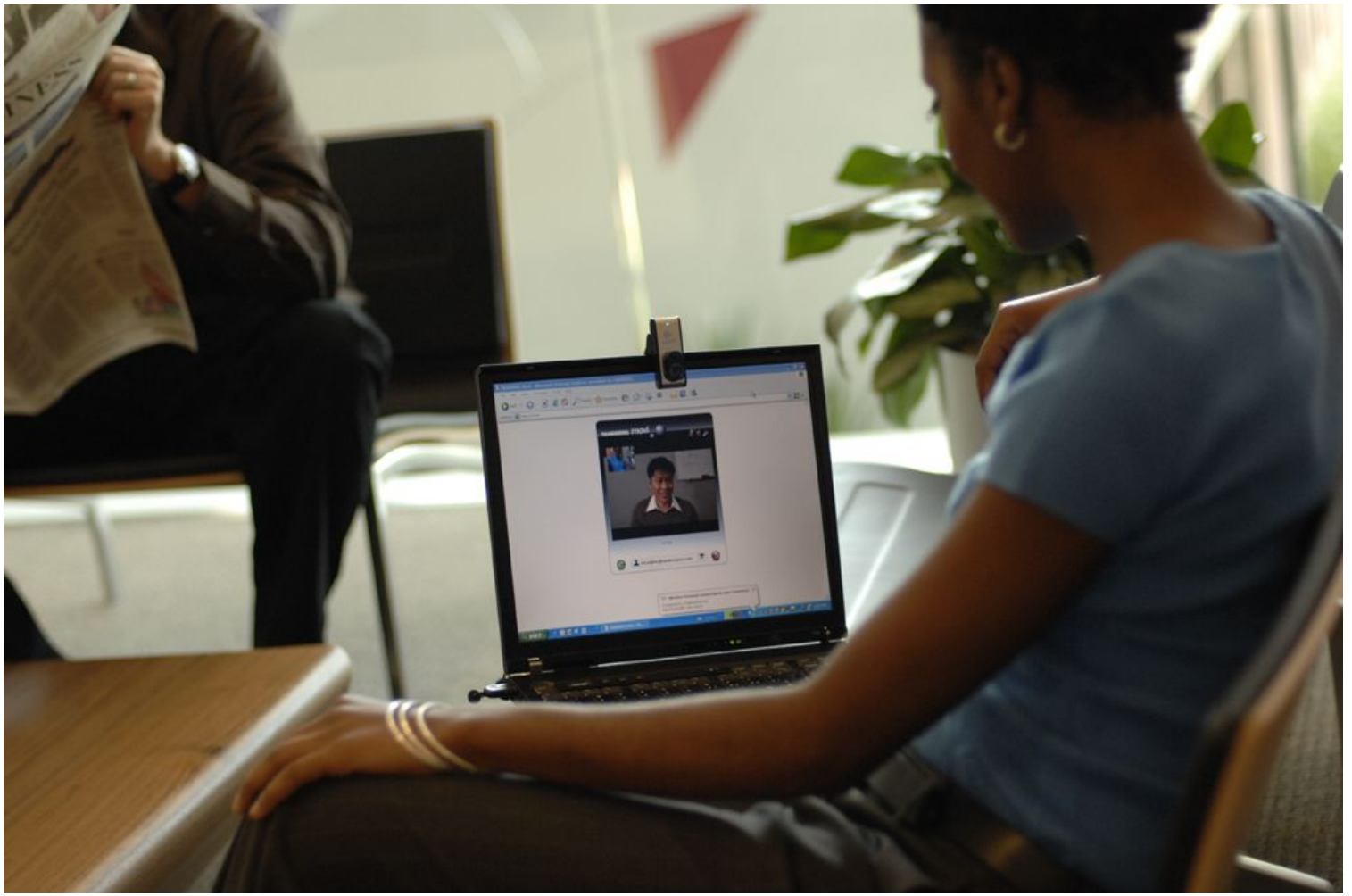
TANDBERG
www.tandberg.com

We develop and sell…

Meeting room systems

Telepresence systems

Personal systems



PC based systems

Networking products

… and a lot of other stuff

~1500 employees worldwide

~400 R&D engineers

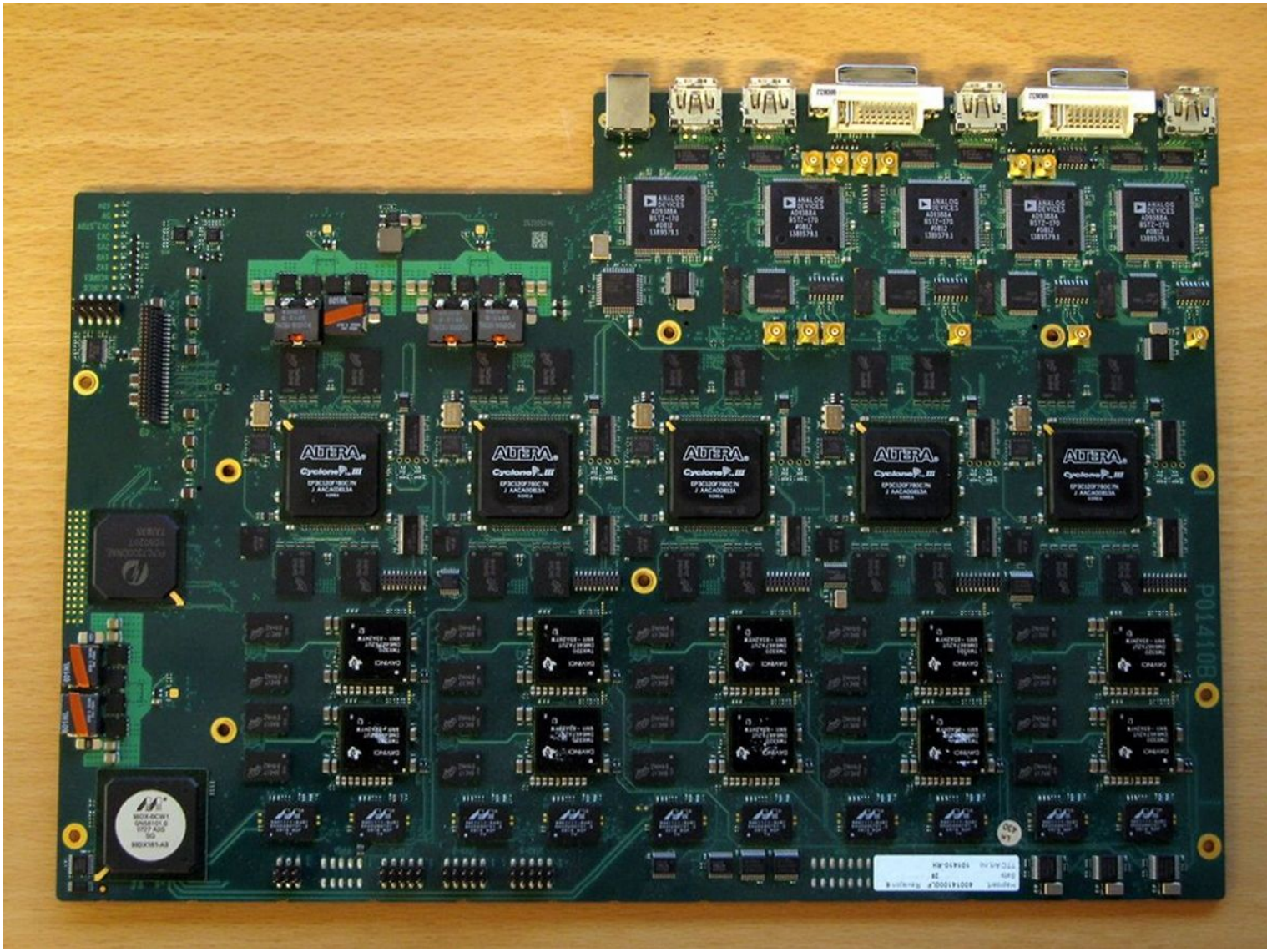
~250 write code every day

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

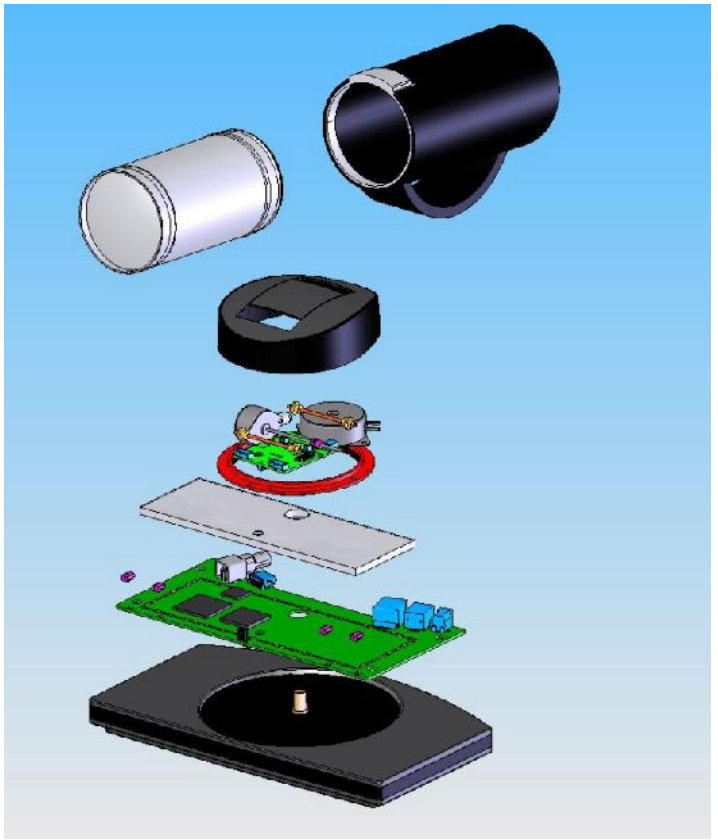
most of us work with software development

but we also do…

Electronics / Hardware



Mechanics



Industrial Design

1992

2007

Looking into

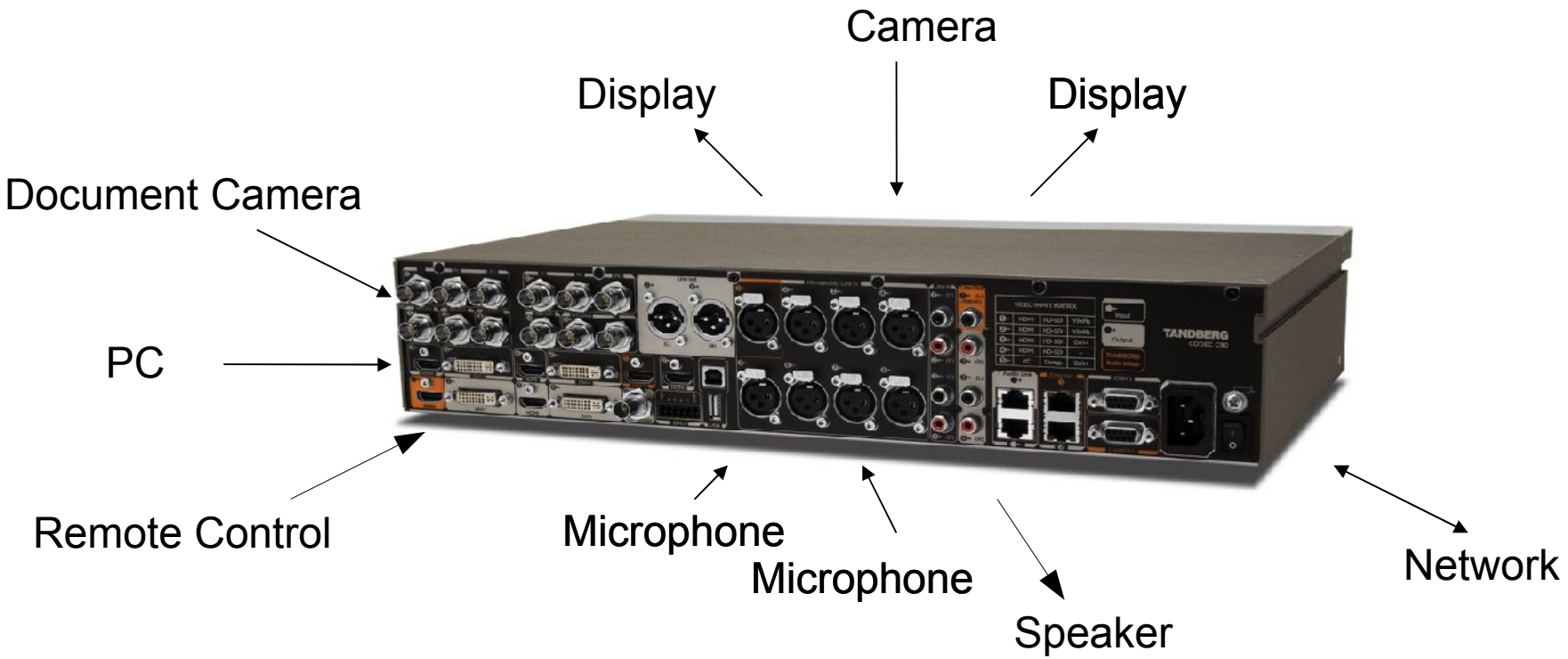
the Future

TANDBERG Codec C90 - “The Saturn Project”


TANDBERG
CODEC C90
VIDEO INPUT MATRIX
Input
Output
TANDBERG basic setup
Line out
Microphone/ Line in
Line In
Line Out
COM 1
GPIO
USB

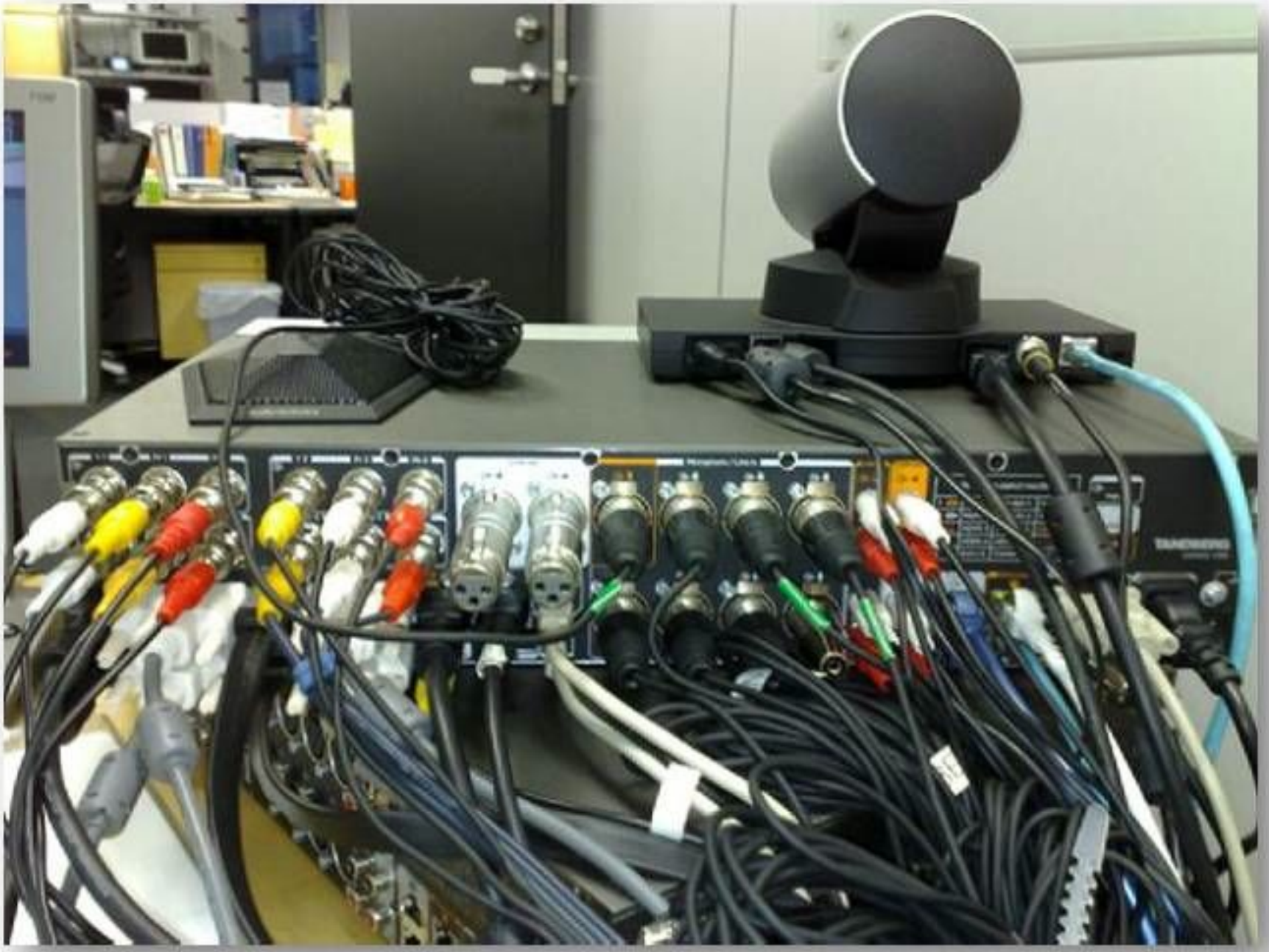
C90 video (1:19)

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



C90 Features:

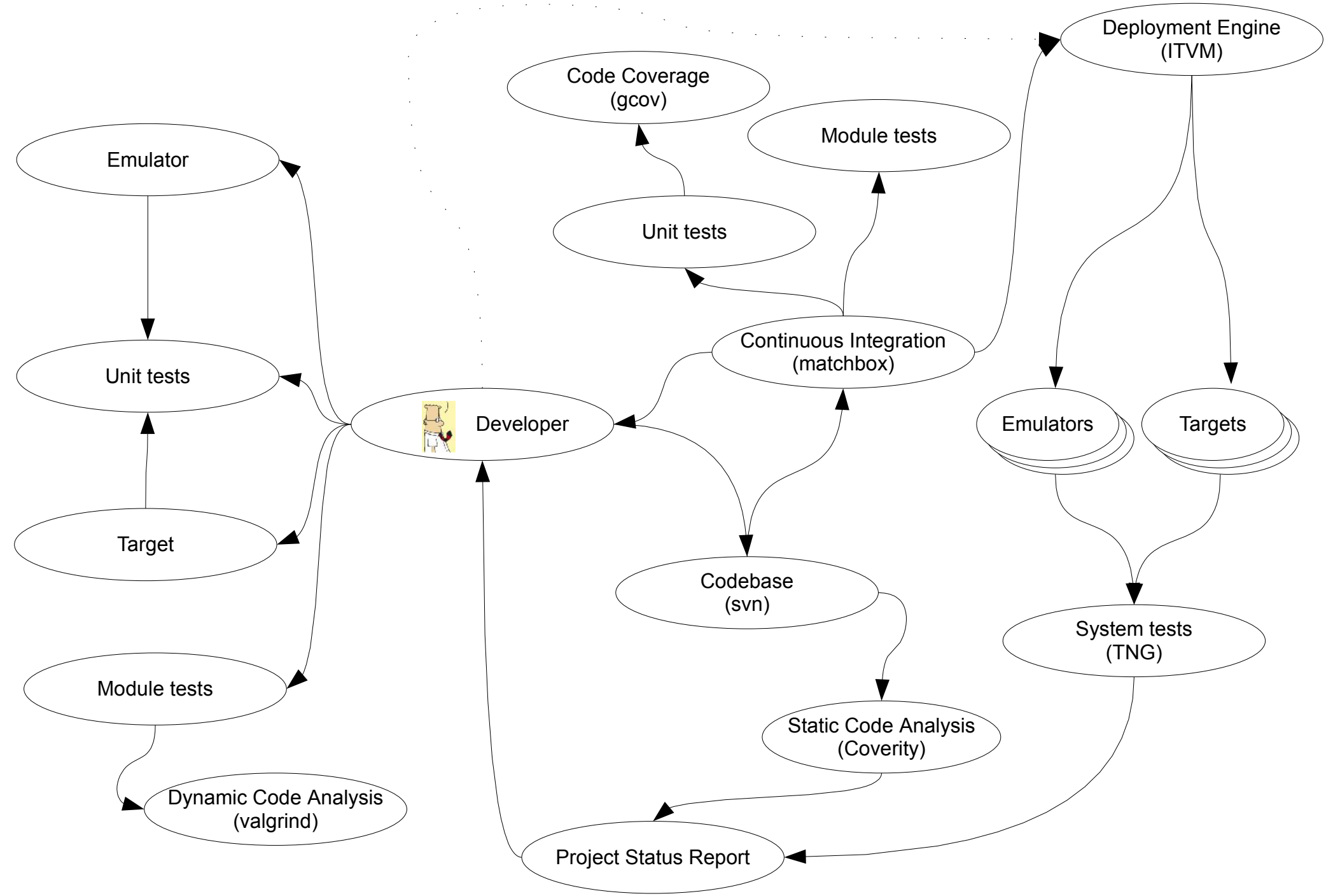
- realtime H.264 encoding/decoding
- full HD 1080p30, (4+4) concurrent streams
- 12 high definition video sources
- 8 high quality audio sources
- support for many-to-many communication
- Interoperability through H323 and SIP
- API for integration and remote control

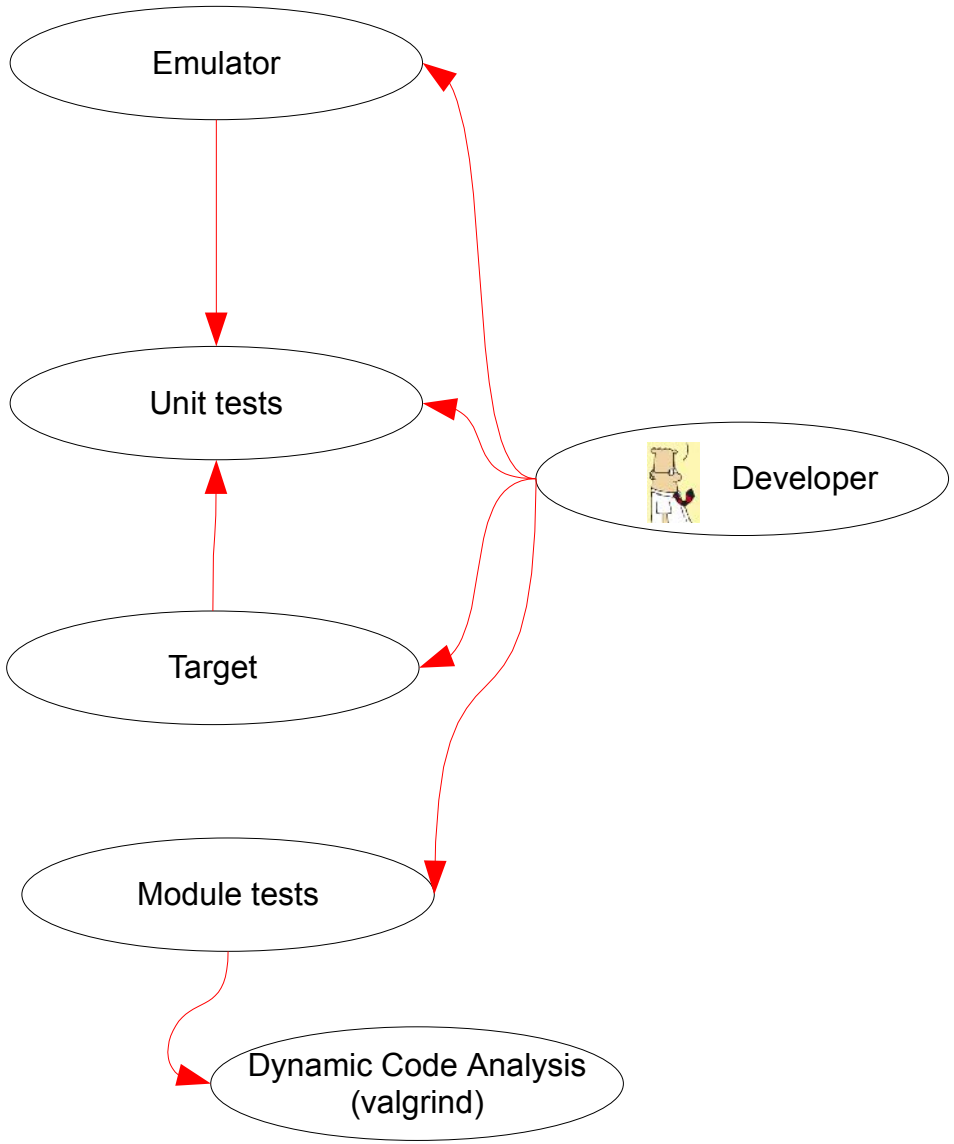


Development Practices in the Saturn project

- 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
- Continuous planning
- Always attack high risks first

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





Developer can work locally and use an emulator or submit an image to a real target (if available).

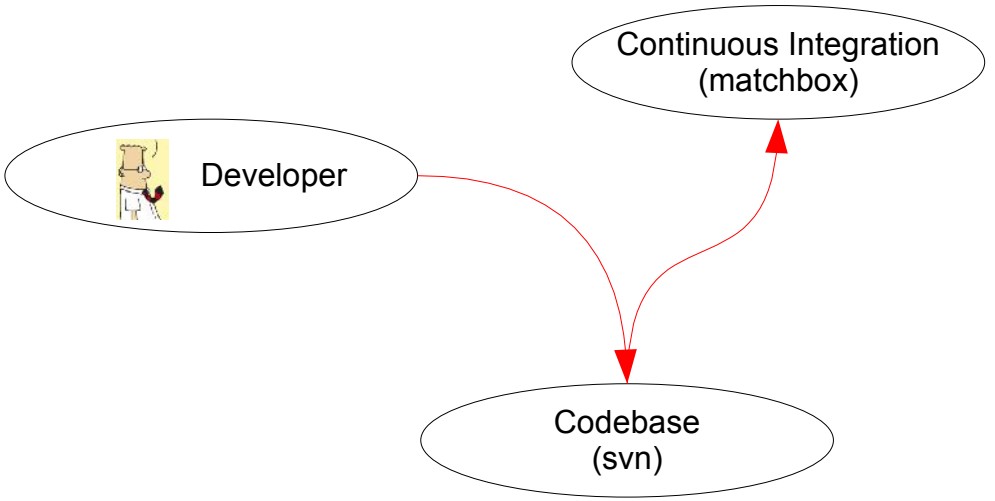
Notice that the unit tests can be executed standalone, but also by the emulator and the target. This is very useful as the developer environment, the emulator and target are running on different type of processors, where even the endianess sometimes differ.

We also have a framework for module testing where we can mock out other modules and work on larger chuncks of software independently of the whole system. This is extremely useful when doing dynamic code analysis for example.

Our build system is very fast, so a developer will typically get feedback within seconds or a few minutes.

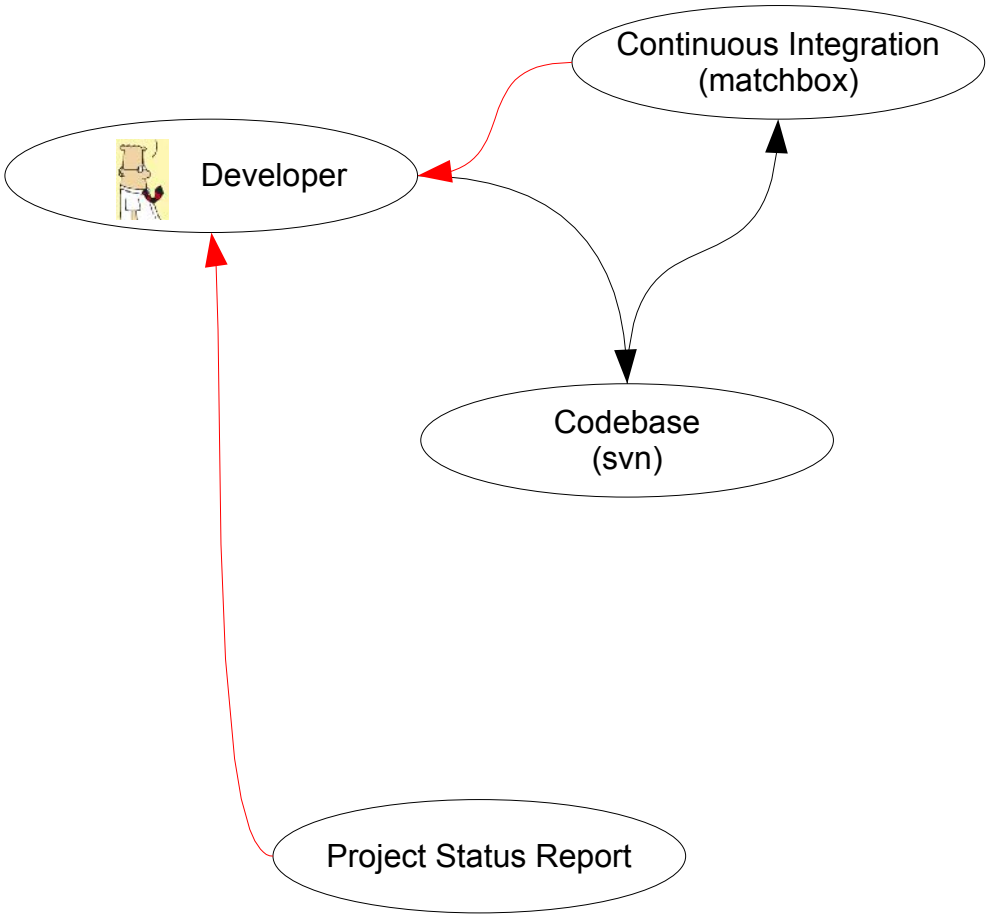
When developer make a change to the shared codebase it is immediately picked up by matchbox, our continuous integration system. Matchbox will build all products depending on that change, perhaps as many as ten products using different compilers targeting various processors.

All compilation is with high warning levels (-Wall, -Wextra) and warnings are treated as errors (-Werror)



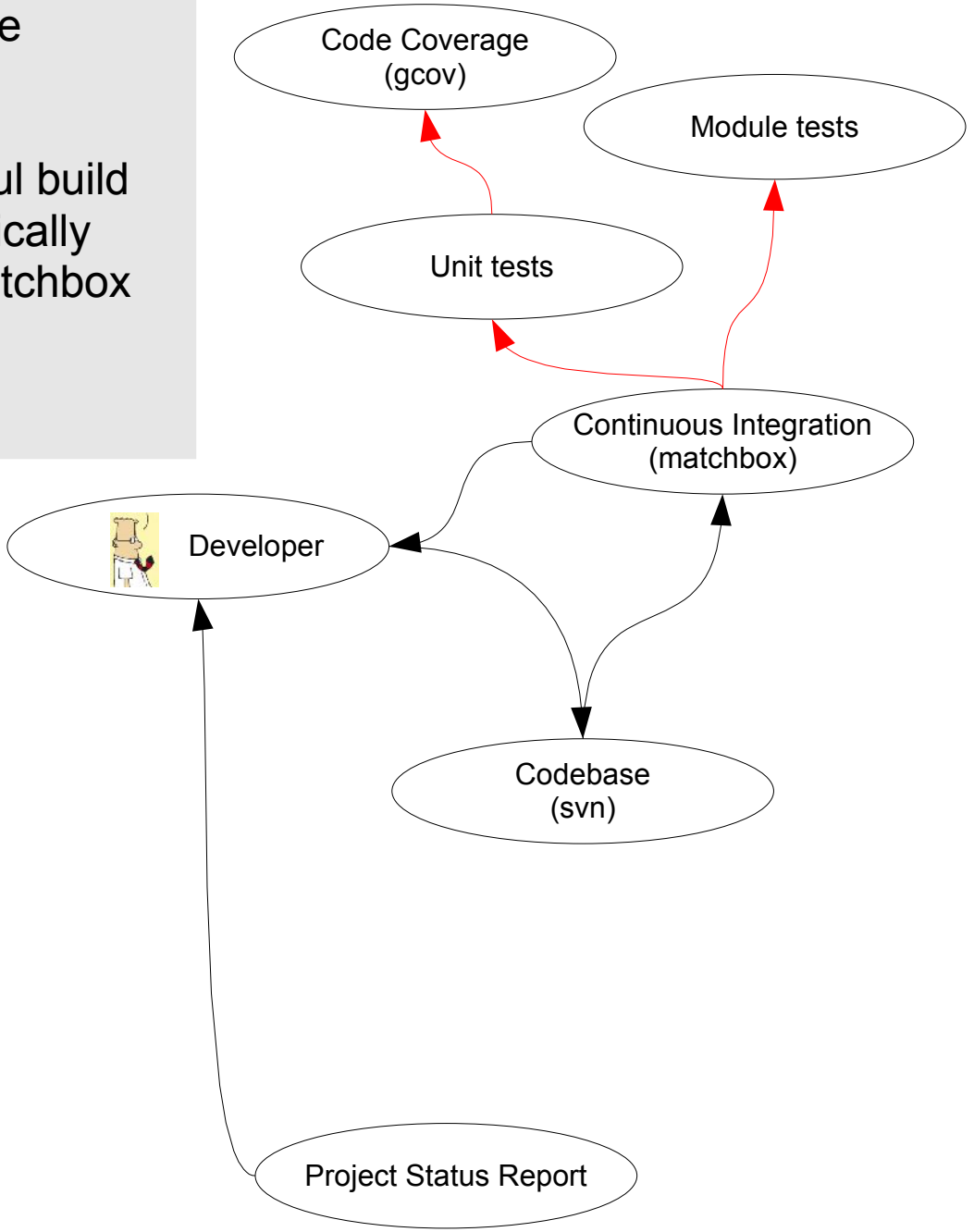
By the way, before checking in, developer will always check both the current build status and the current project status report.

IMPORTANT: Do not check in new stuff on red. Do not build stuff on top of garbage.



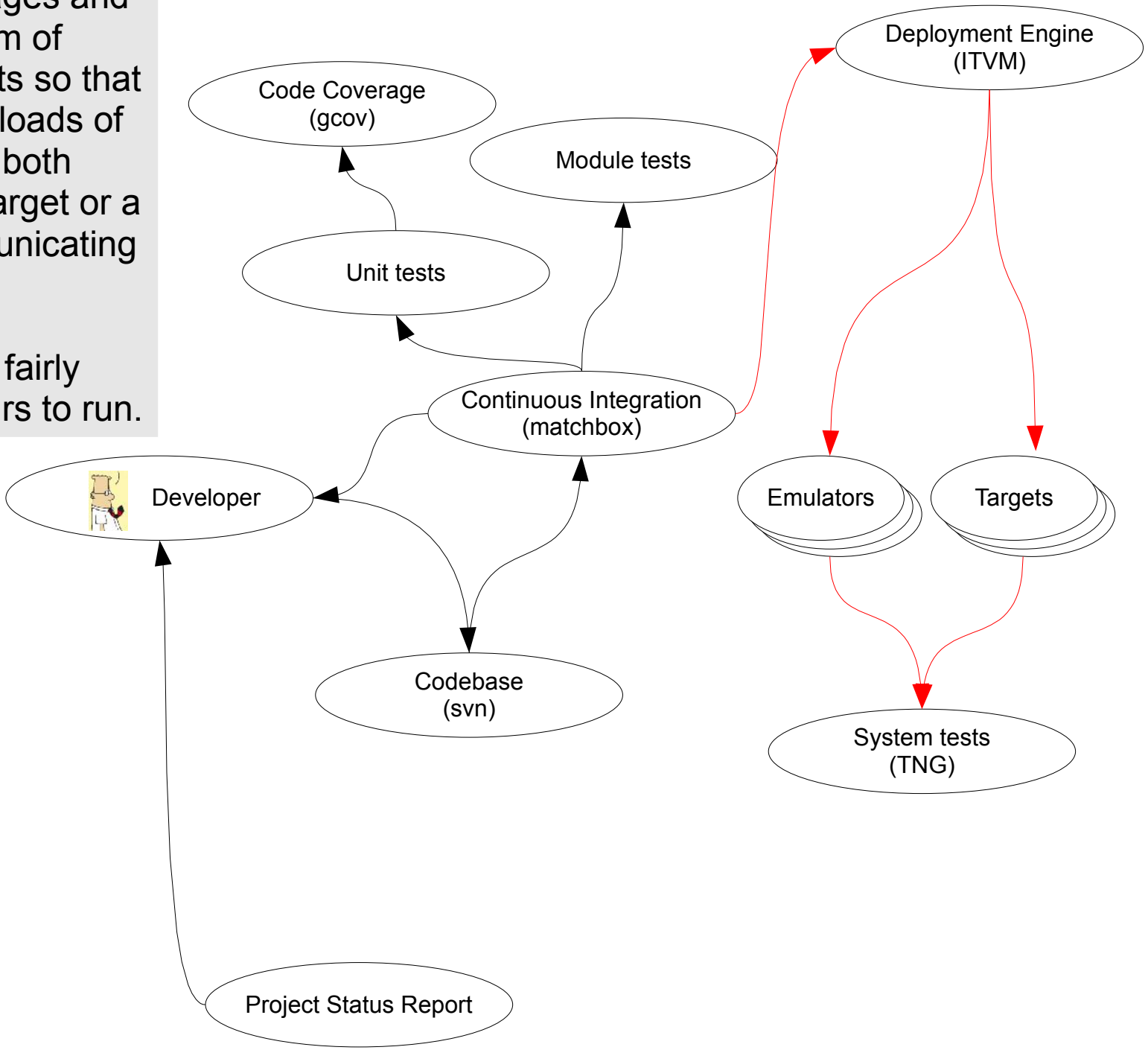
Matchbox will also run all unit tests, measure code coverage and run module tests for all products affected by the change.

We have many powerful build servers. Developer typically gets feedback from matchbox within a few minutes.



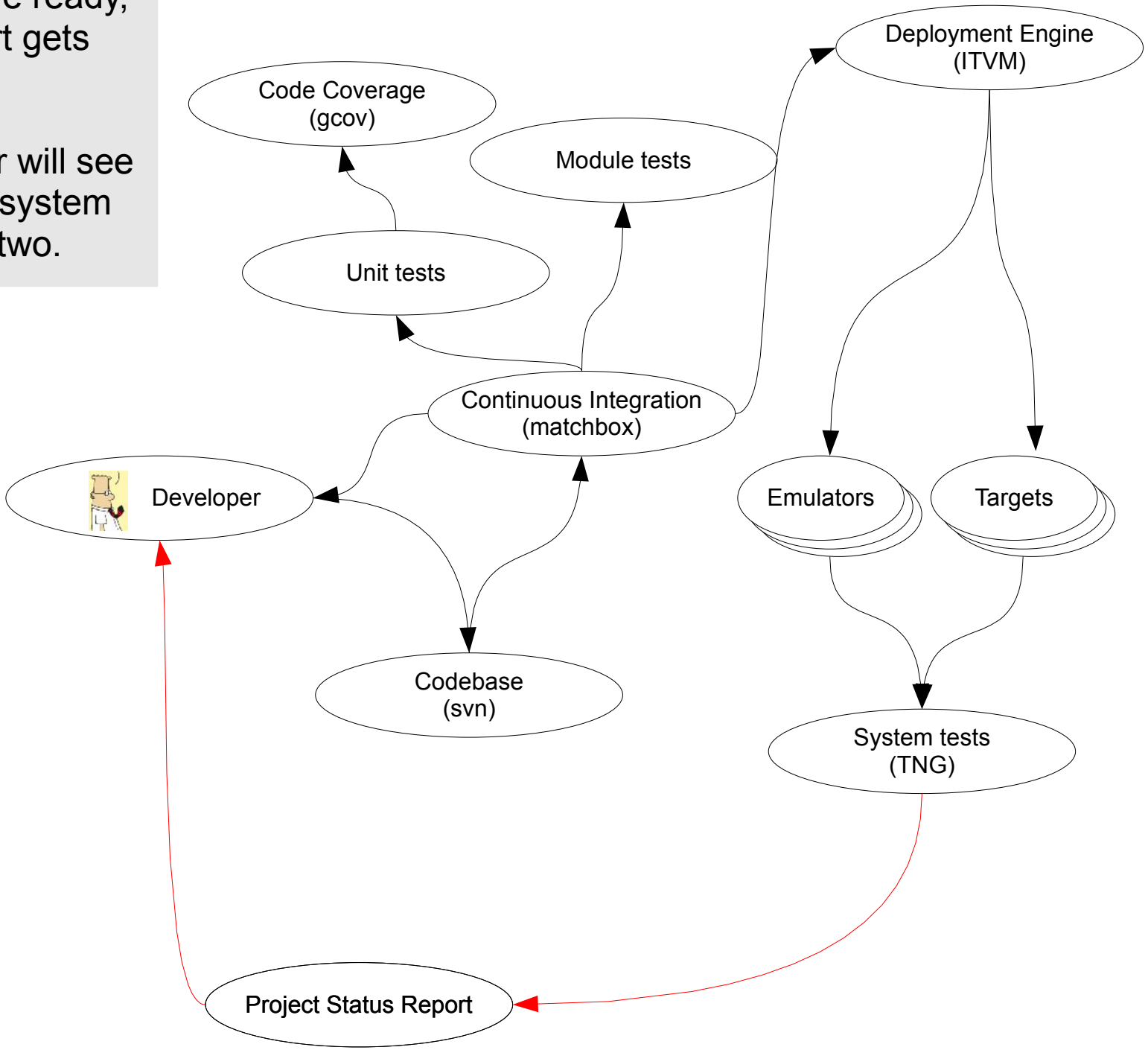
If our matchbox is happy, then ITVM take the built images and deploy them onto a farm of emulators and/or targets so that TNG can start running loads of scripted test scenarios both testing one particular target or a group of targets communicating with each other.

Some of the tests runs fairly quickly other takes hours to run.

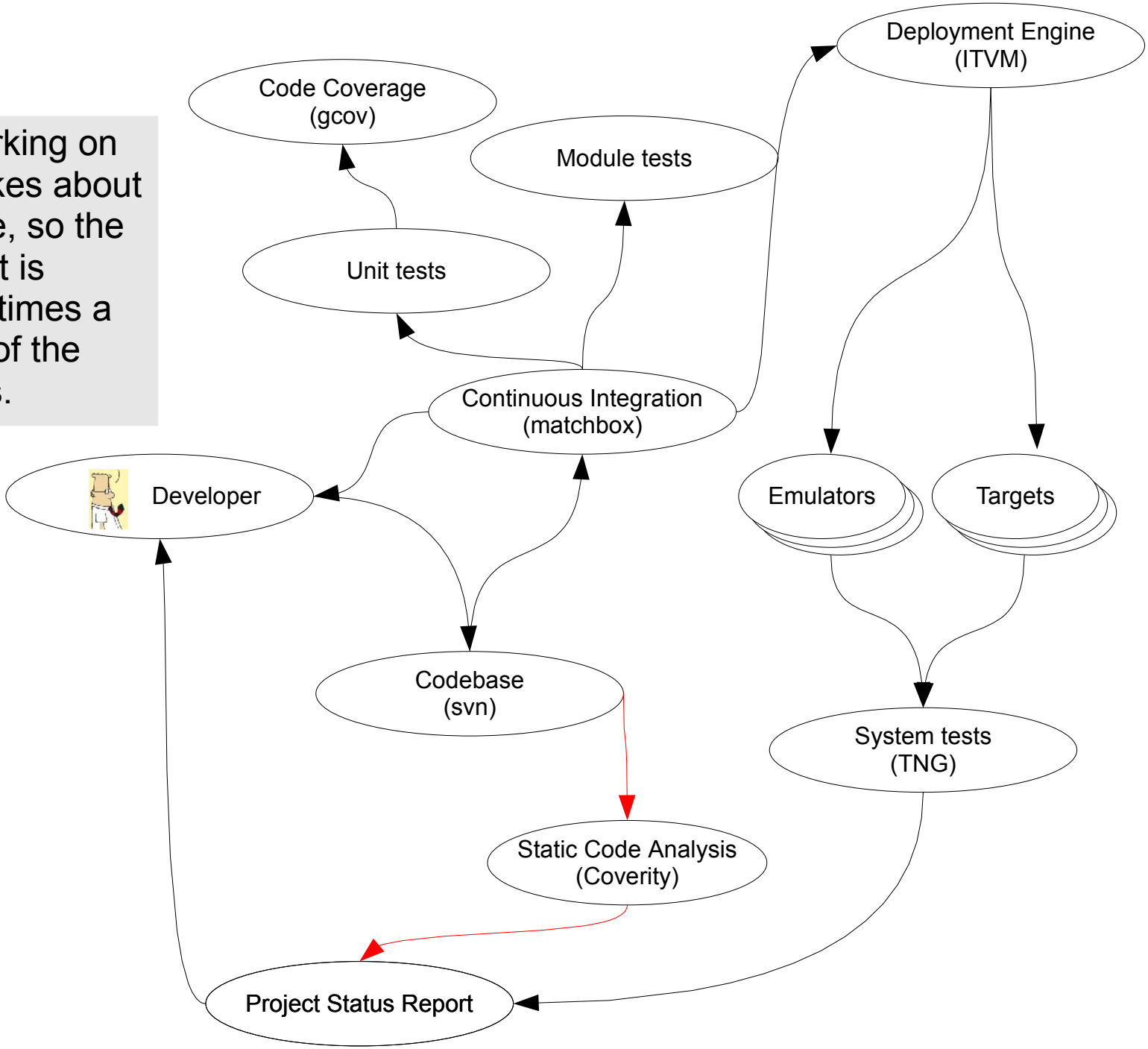


As soon as the results from various system tests are ready, the project status report gets updated.

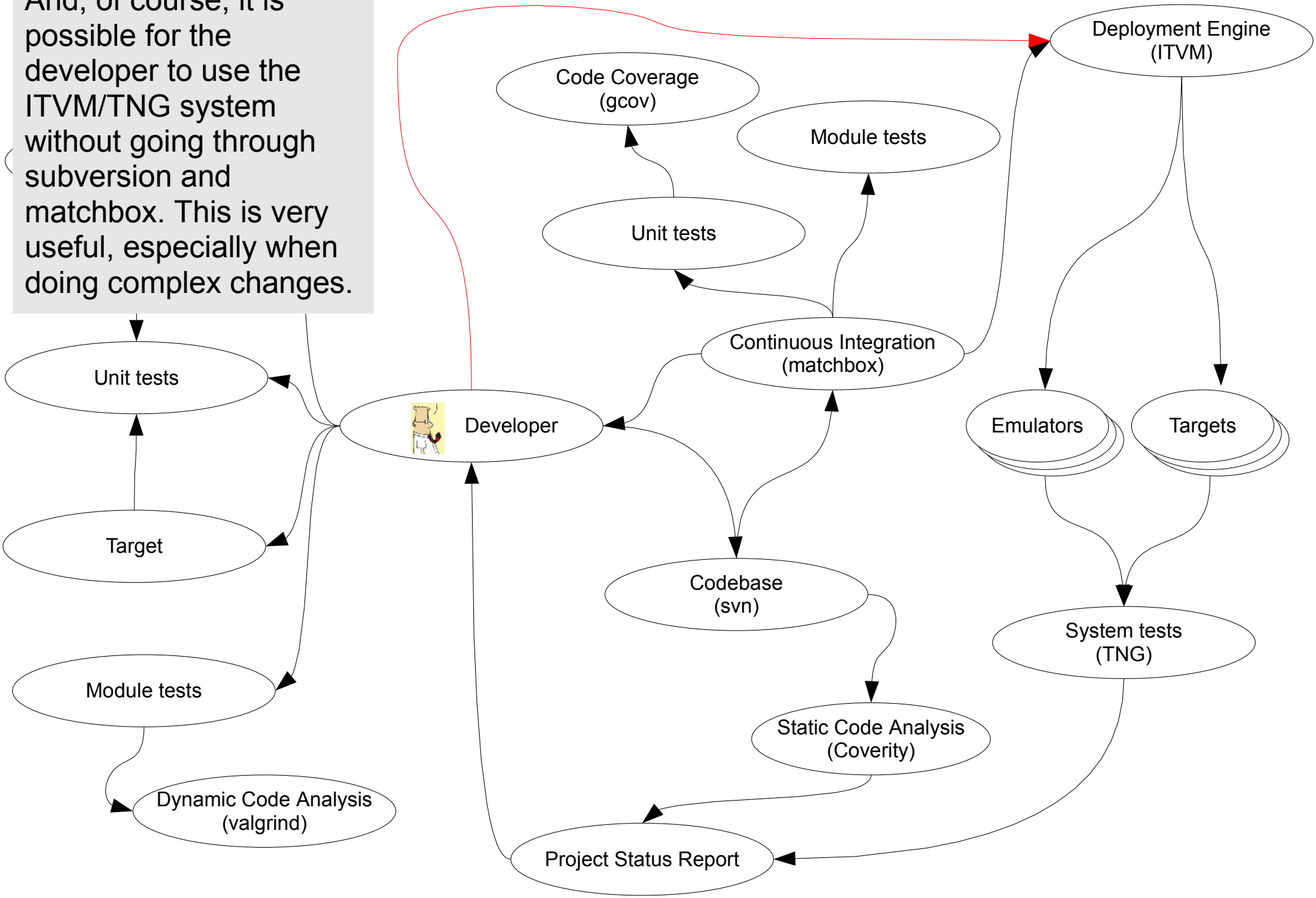
Typically, the developer will see the results from all the system tests within an hour or two.



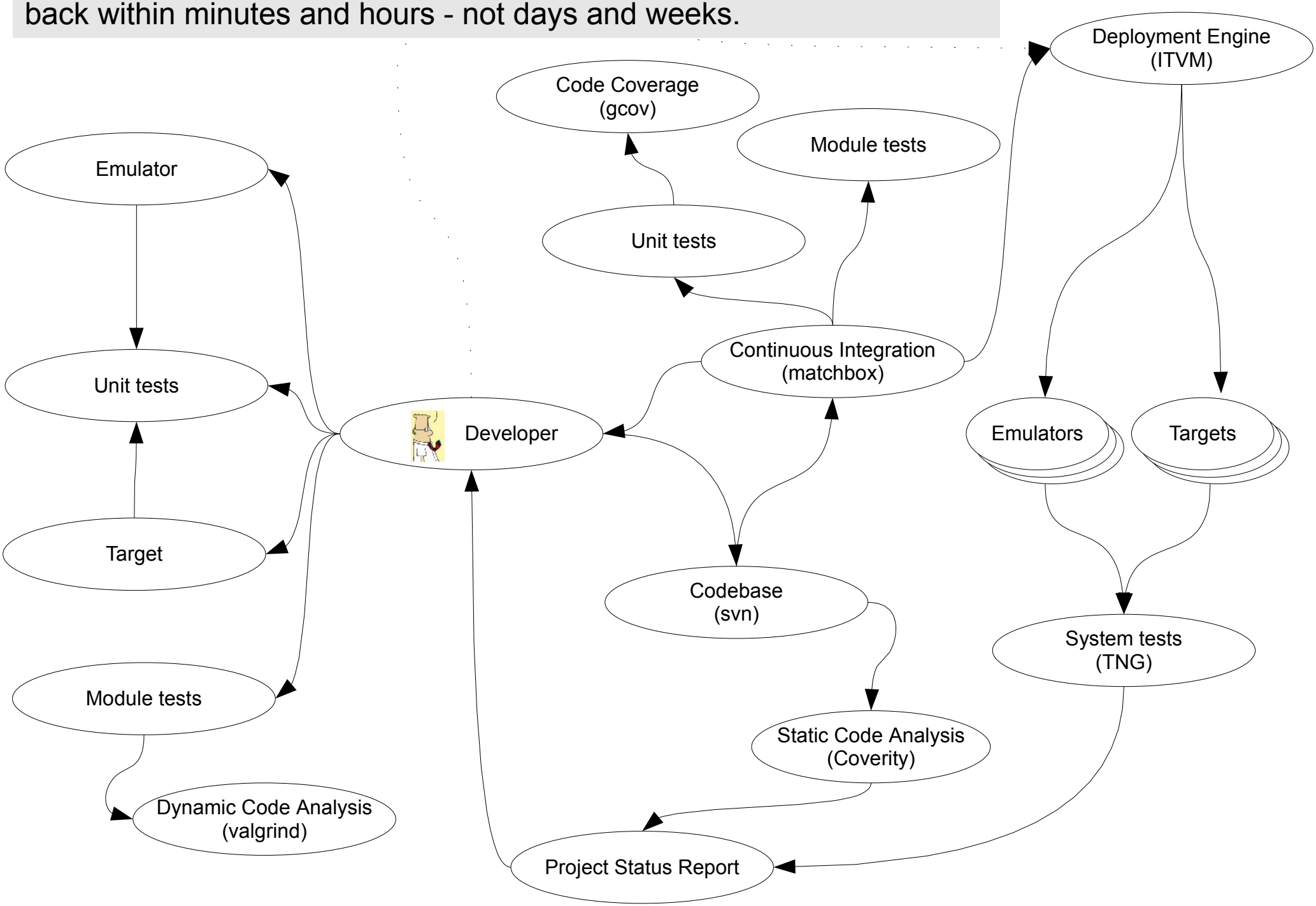
Coverity is also working on the codebase. It takes about 12 hours to execute, so the project status report is updated about two times a day with the result of the static code analysis.



And, of course, it is possible for the developer to use the ITVM/TNG system without going through subversion and matchbox. This is very useful, especially when doing complex changes.



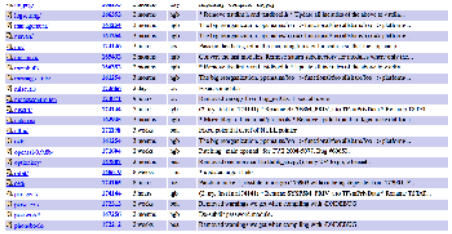
This workflow represents the feedback mechanism that developers rely on when applying changes to the codebase. Bad changes are reported back within minutes and hours - not days and weeks.



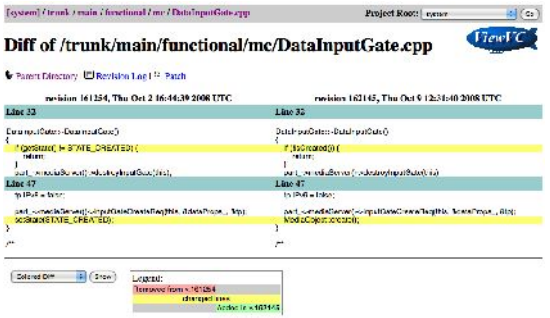
Other aspects of the Saturn project:

- 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)

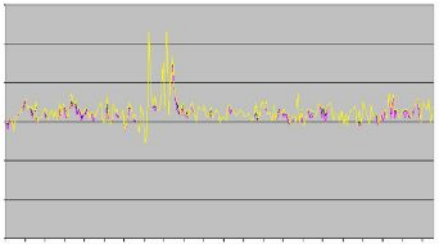
Example of visual feedback (HTML pages used by all/most developers)



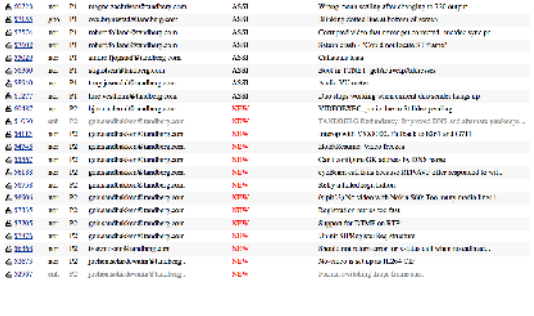
viewvc



diff from viewvc

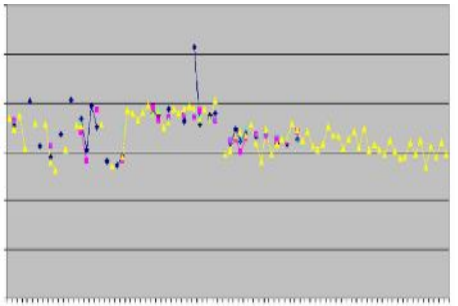


audio delay trend

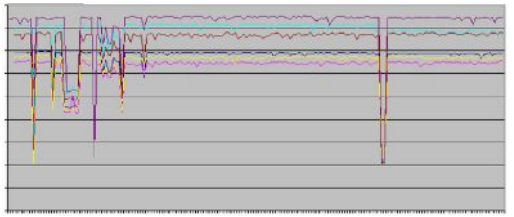


bugzilla

irc channel

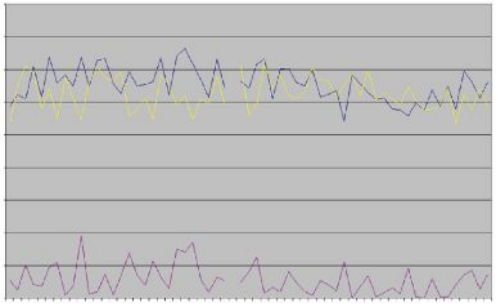


H.264 delay trend

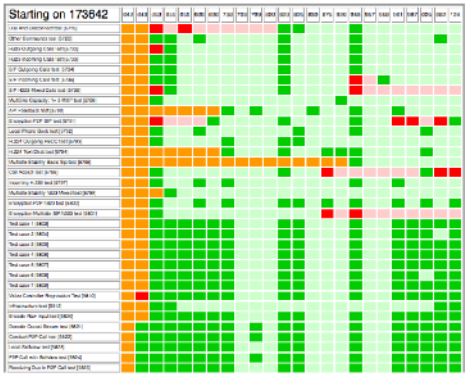


PESQ trend

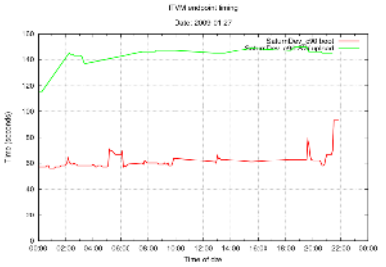
continuous integration



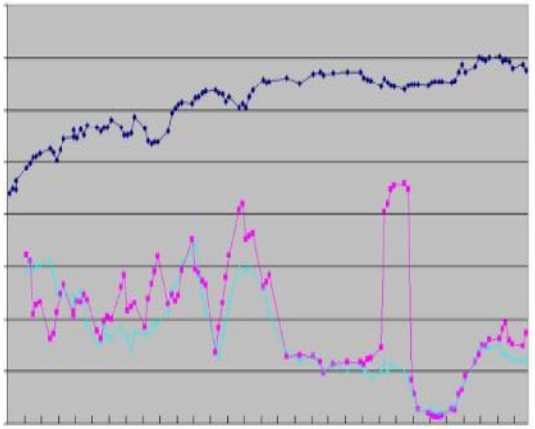
lipsync trend



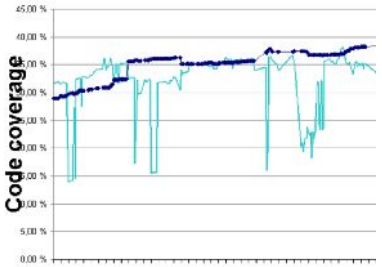
system tests

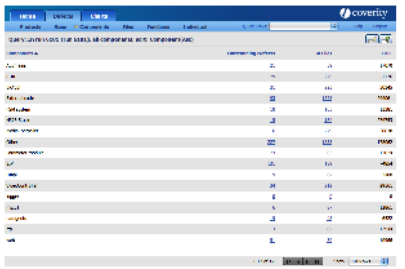


endpoint timing

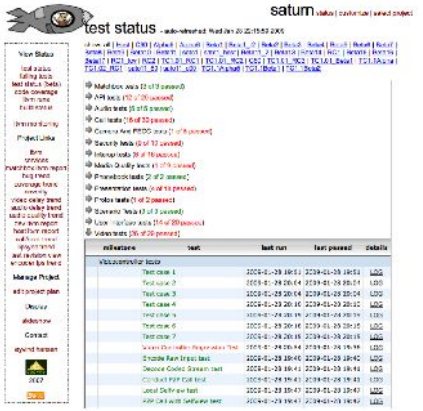


defect trends





coverity



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.

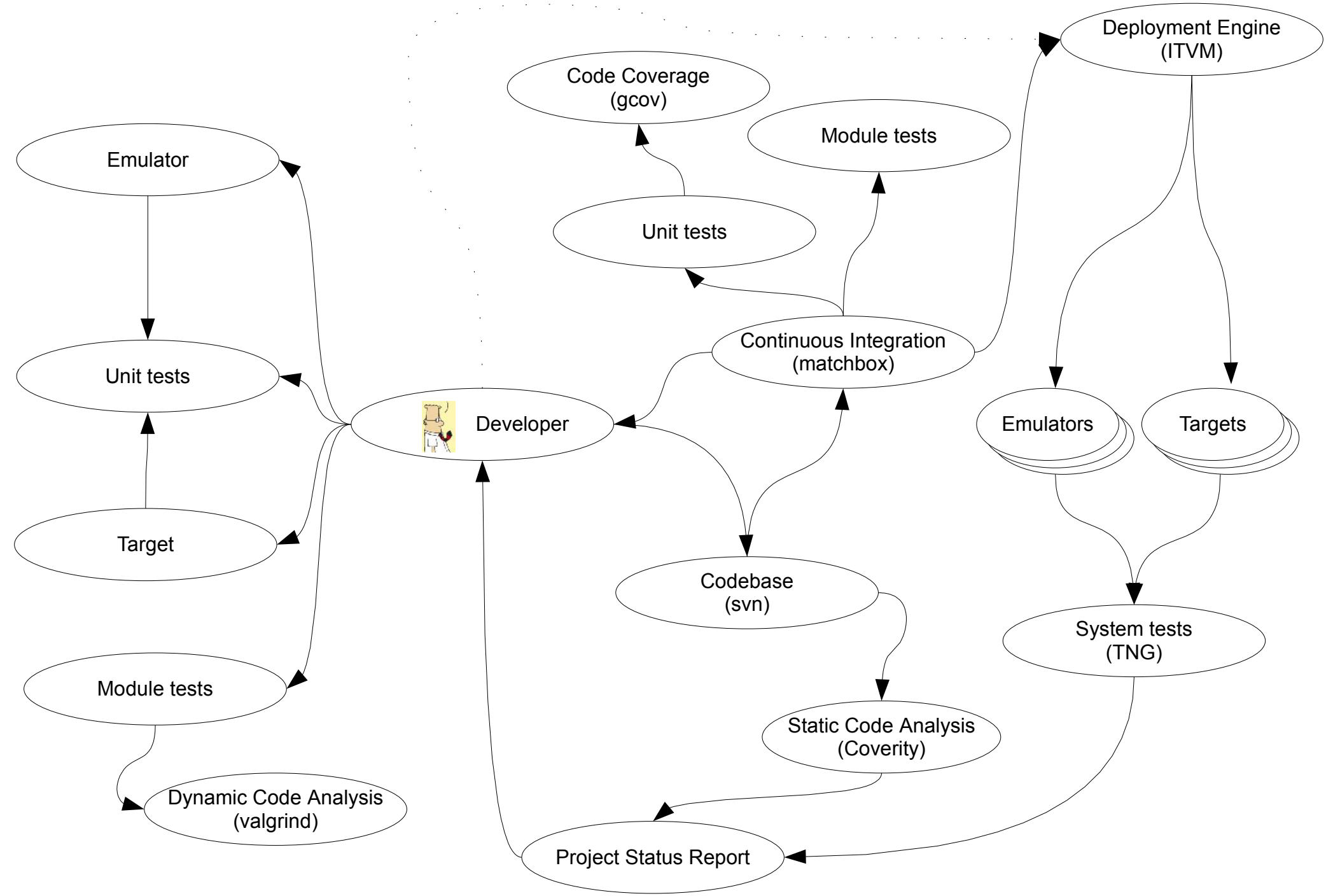
We believe that software development is a continuous learning process and a cooperative game of communication between professionals. Software development is about repeating cycles of preparing, changing, observing, reflecting, and learning.

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...

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



Product Development in Tandberg

Product Development in Tandberg

- We spend very little energy on things that are not essential

Product Development in Tandberg

- We spend very little energy on things that are not essential
- "Plans are nothing, planning is everything"

Product Development in Tandberg

- We spend very little energy on things that are not essential
- "Plans are nothing, planning is everything"
- We do not write hours, we do not measure project cost

Product Development in Tandberg

- We spend very little energy on things that are not essential
- "Plans are nothing, planning is everything"
- We do not write hours, we do not measure project cost
- Decisions are delayed as much as possible

Product Development in Tandberg

- We spend very little energy on things that are not essential
- "Plans are nothing, planning is everything"
- We do not write hours, we do not measure project cost
- Decisions are delayed as much as possible
- To fail is OK, therefore we deliver spectacular stuff

Product Development in Tandberg

- We spend very little energy on things that are not essential
- "Plans are nothing, planning is everything"
- We do not write hours, we do not measure project cost
- Decisions are delayed as much as possible
- To fail is OK, therefore we deliver spectacular stuff
- Doers are very much respected in Tandberg

Product Development in Tandberg

- We spend very little energy on things that are not essential
- "Plans are nothing, planning is everything"
- We do not write hours, we do not measure project cost
- Decisions are delayed 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

Product Development in Tandberg

- We spend very little energy on things that are not essential
- "Plans are nothing, planning is everything"
- We do not write hours, we do not measure project cost
- Decisions are delayed 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

Product Development in Tandberg

- We spend very little energy on things that are not essential
- "Plans are nothing, planning is everything"
- We do not write hours, we do not measure project cost
- Decisions are delayed 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”

Product Development in Tandberg

- We spend very little energy on things that are not essential
- "Plans are nothing, planning is everything"
- We do not write hours, we do not measure project cost
- Decisions are delayed 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

Product Development in Tandberg

- We spend very little energy on things that are not essential
- "Plans are nothing, planning is everything"
- We do not write hours, we do not measure project cost
- Decisions are delayed 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
- Little documentation gives effective communication

Product Development in Tandberg

- We spend very little energy on things that are not essential
- "Plans are nothing, planning is everything"
- We do not write hours, we do not measure project cost
- Decisions are delayed 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
- Little documentation gives effective communication
- Slack is embedded, and “skunk work” projects appreciated

Product Development in Tandberg

- We spend very little energy on things that are not essential
- "Plans are nothing, planning is everything"
- We do not write hours, we do not measure project cost
- Decisions are delayed 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
- Little documentation gives effective communication
- Slack is embedded, and “skunk work” projects appreciated
- The company builds on trust

Product Development in Tandberg

- We spend very little energy on things that are not essential
- "Plans are nothing, planning is everything"
- We do not write hours, we do not measure project cost
- Decisions are delayed 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
- Little documentation gives effective communication
- Slack is embedded, and “skunk work” projects appreciated
- The company builds on trust
- Developers are treated as professionals, not as resources

Product Development in Tandberg

- We spend very little energy on things that are not essential
- "Plans are nothing, planning is everything"
- We do not write hours, we do not measure project cost
- Decisions are delayed 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
- Little documentation gives effective communication
- Slack is embedded, and “skunk work” projects appreciated
- The company builds on trust
- Developers are treated as professionals, not as resources
- Fun and Profit

Product Development in Tandberg

- We spend very little energy on things that are not essential
- "Plans are nothing, planning is everything"
- We do not write hours, we do not measure project cost
- Decisions are delayed 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
- Little documentation gives effective communication
- Slack is embedded, and “skunk work” projects appreciated
- The company builds on trust
- Developers are treated as professionals, not as resources
- Fun and Profit

We follow principles, not processes!

!

Appendix

It takes a lot of effort to stay ahead of 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

…

Hackers Corner – internal seminars and studygrups

Inhouse conferences

TechZone Lillehammer 2007

~ 190 engineers, 32 talks, 4 tracks

TechZone Lysaker 2007

~ 260 engineers, 40 talks, 5 tracks

TechZone Barcelona 2008

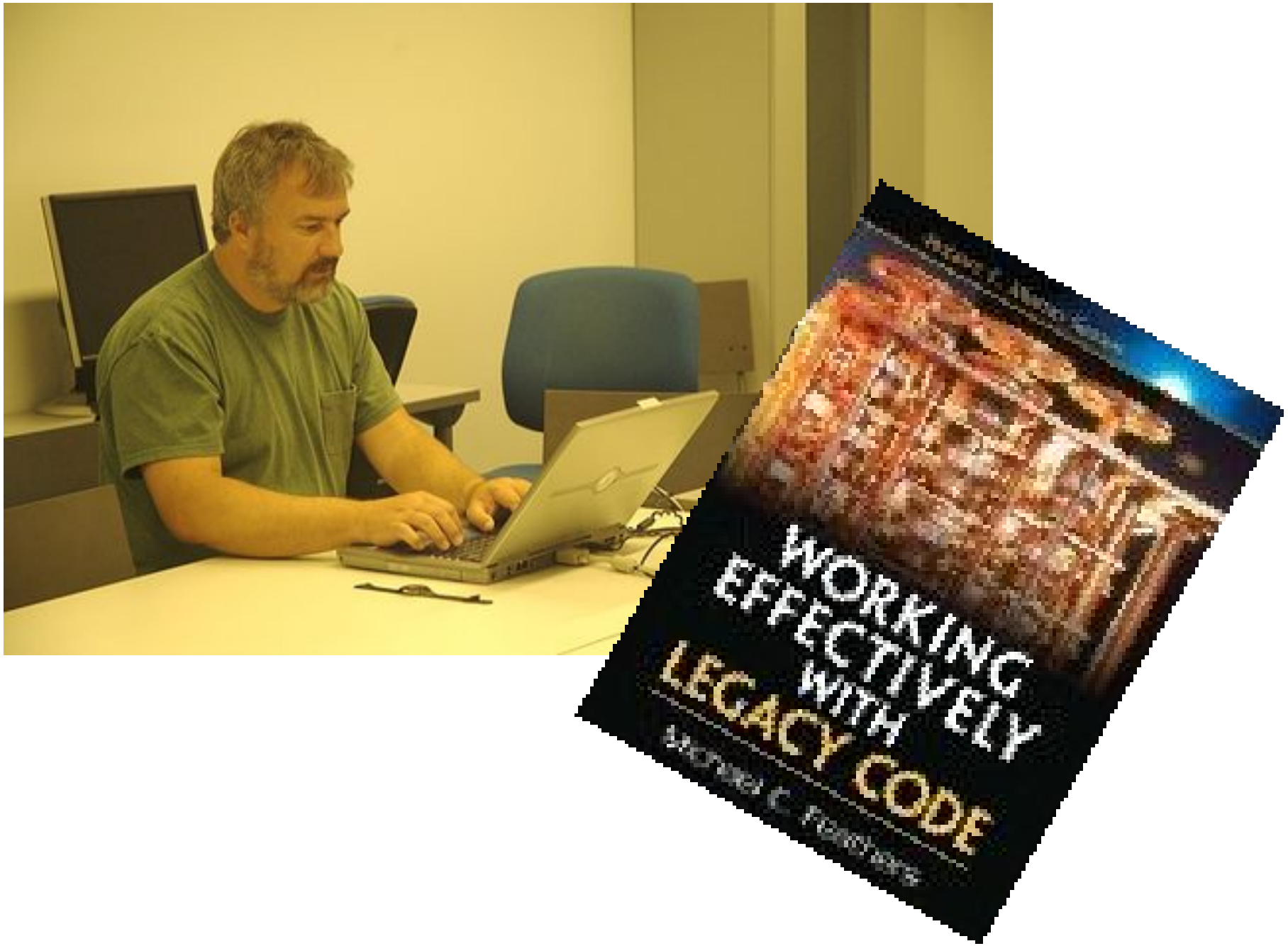
~ 380 engineers, 43 talks, 5 tracks

engage industry experts…

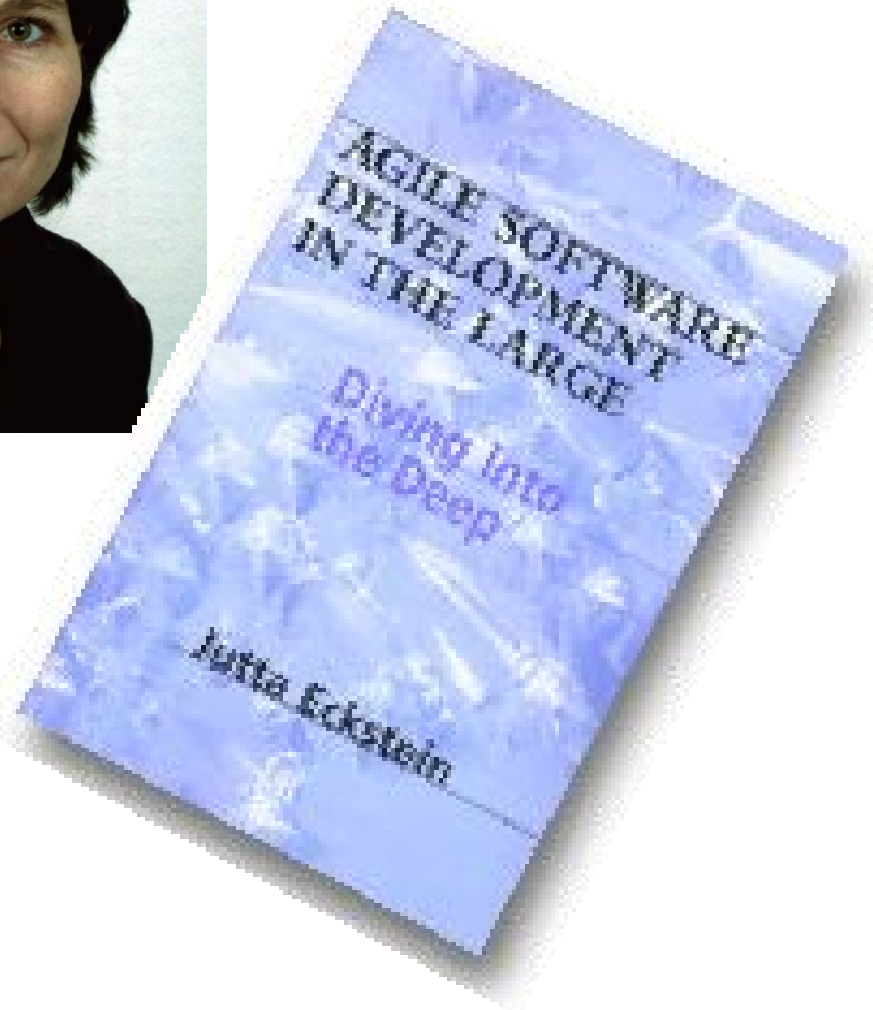
Dana Bredemeyer - Software Architecture

Kevlin Henney - C++, C, UML, Pattern-Based Software Development

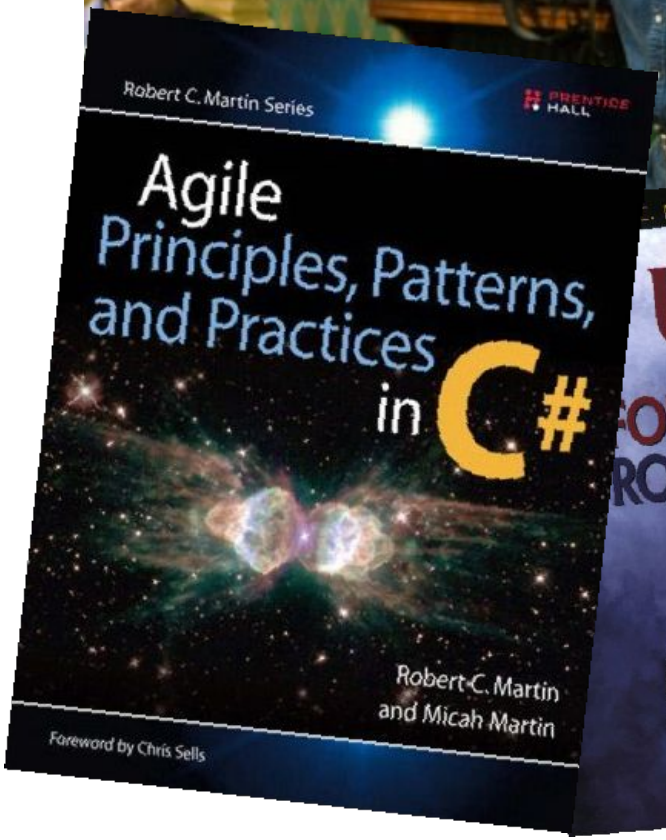
Michael Feathers - Pair-Programming, Testing, Professionalism

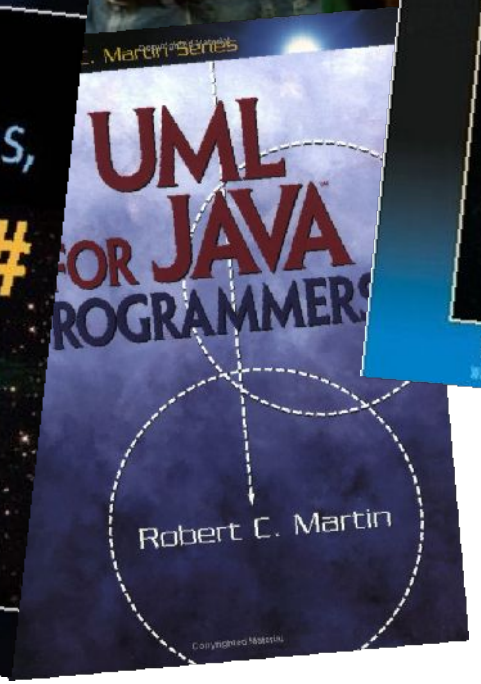


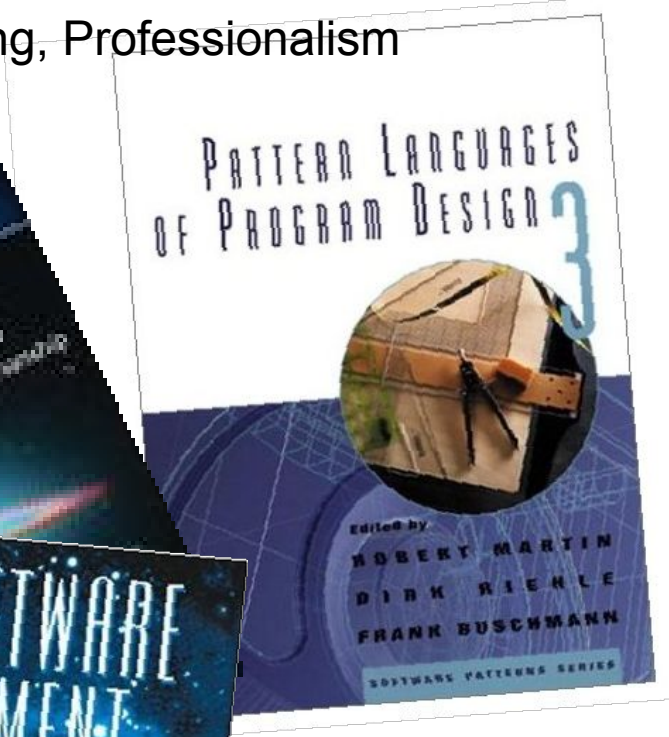
Jutta Eckstein - Project Management, Agile Planning, Retrospectives

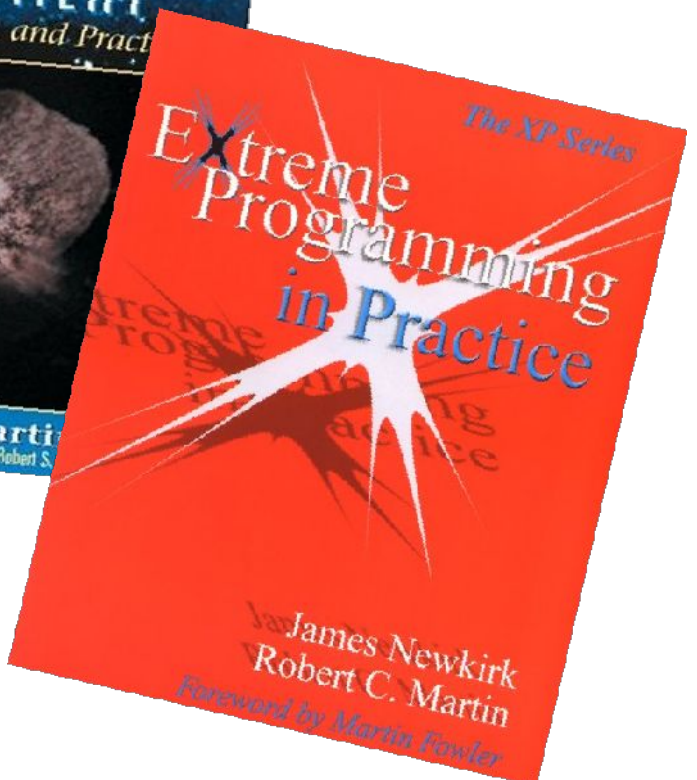


Robert C. Martin “Uncle Bob” - TDD, Pair Programming, Professionalism

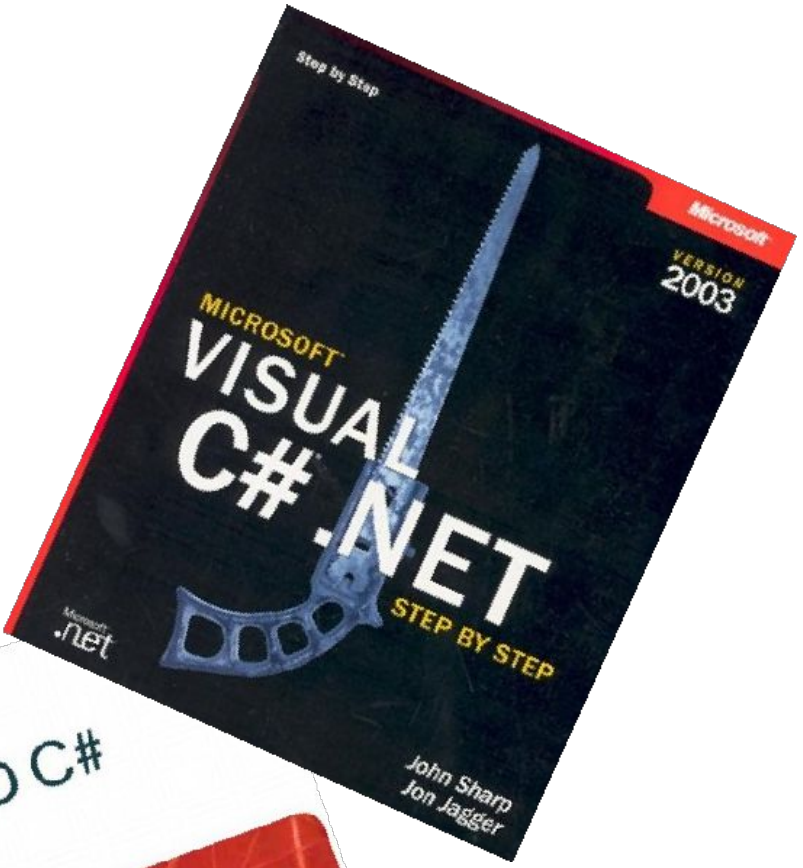




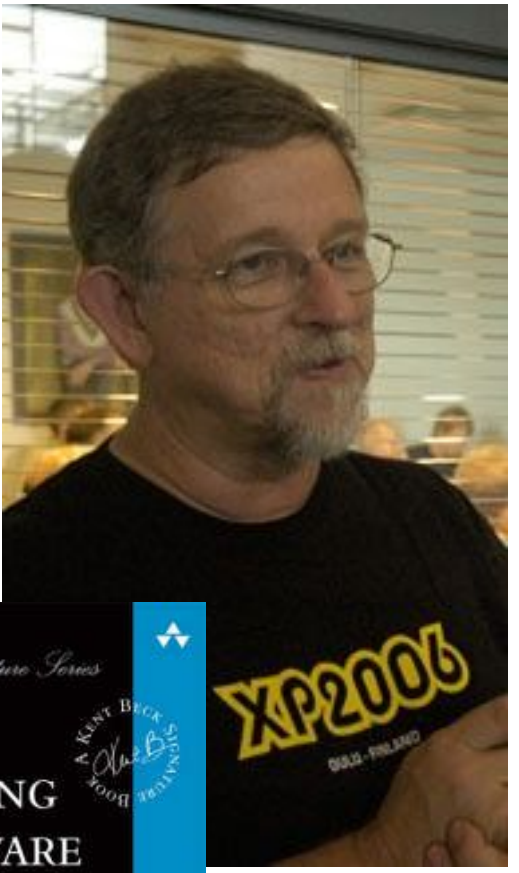


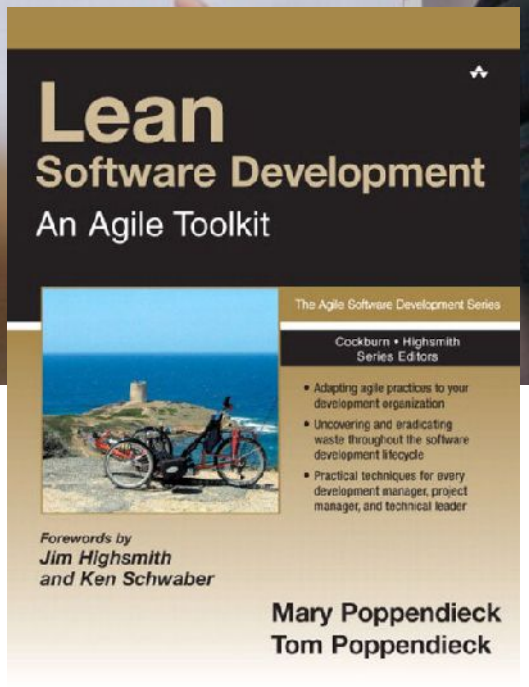


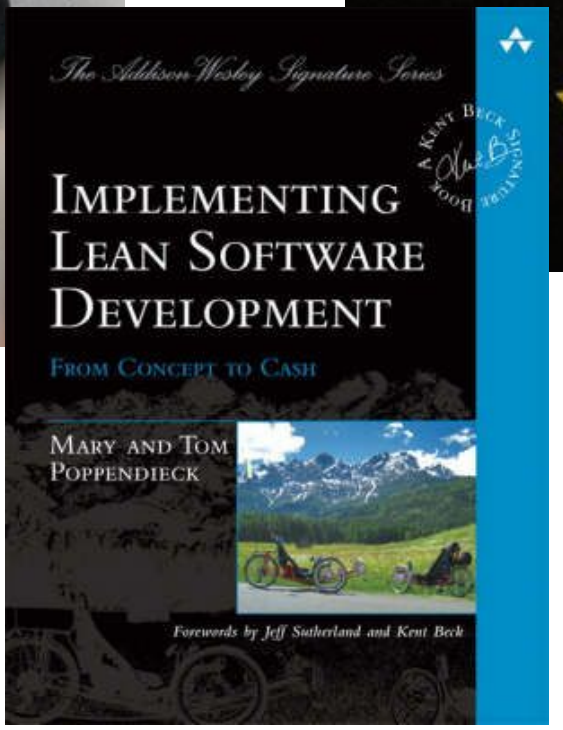
Jon Jagger - C, C++, UML, Patterns, Agile, TDD



Tom and Mary Poppendieck - Project Management, Lean Software Development







even RMS has been visiting us.

“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)

“I have been working with software development groups all around the world, and you are way ahead of most. Actually this is one of the few companies that I would consider working for.”

(a consultant visiting our R&D department)