

Santa Barbara 2009

# The ABINIT project : software engineering techniques meet simulation of materials and nanosystems



**X. Gonze**  
**Université Catholique de Louvain,**  
**Louvain-la-neuve, Belgium**

# Outline

- I. ABINIT community ... the bazaar
- II. Software engineering techniques ... hold everything
- III. Spin-orbit coupling and dynamical properties of solids

ABINIT : first-principle approach to material and nanosystem properties.  
Computer Physics Comm.,180, 2582-2615 (2009). 40th Anniversary issue.  
(34 pages, 33 authors)

# ABINIT software project

Ideas (1997) :

- 1) Softwares for first-principles simulations becomes more and more sophisticated : need worldwide collaboration, of specialized, complementary, groups
- 2) Linux software development : ‘free software’ model

Now (2009) :

- 500 kLines of F90
- 80 contributors to ABINITv5
- 1300 people on the mailing list
- also, associated software applications, and their communities

<http://www.abinit.org>

# Overview of ABINITv6

## Methodologies

Density Functional Theory (+U) + Density Functional Perturbation Theory +  
Many-body Perturbation Theory + TDDFT + DMFT

Basis sets (originates from the solid state community ...)

Pseudopotentials/Plane Waves + Projector Augmented Waves + Wavelets

## Capabilities

Total energy, density, accurate electronic structure, geometry optimisation,  
molecular dynamics, vibrational properties, electric field (finite or  
infinitesimal), thermodynamical properties, optical response ...

But also : piezoelectricity, electric field gradients, positron lifetime, Raman  
cross-section, XANES, ...

+ Experimental features ... ABINIT developers have tried many things ...

# GW : scQPGW, parallelism, extrapolar

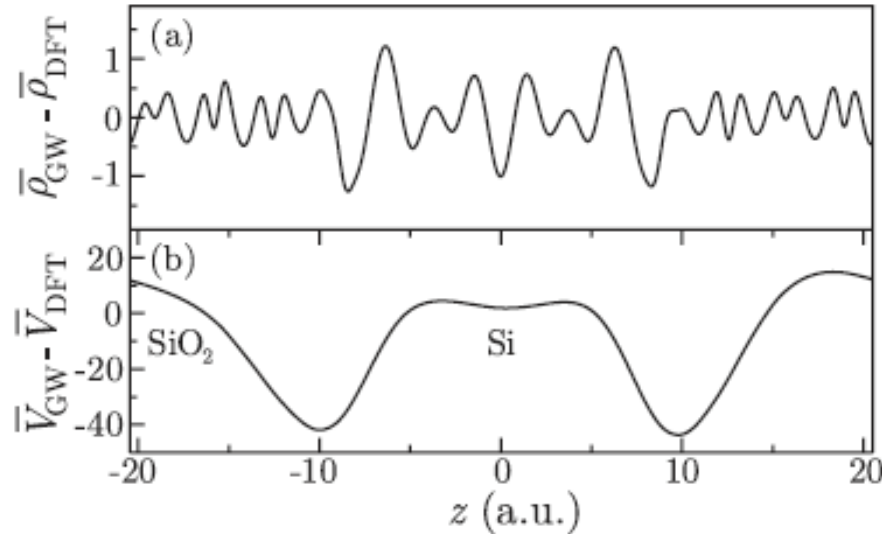
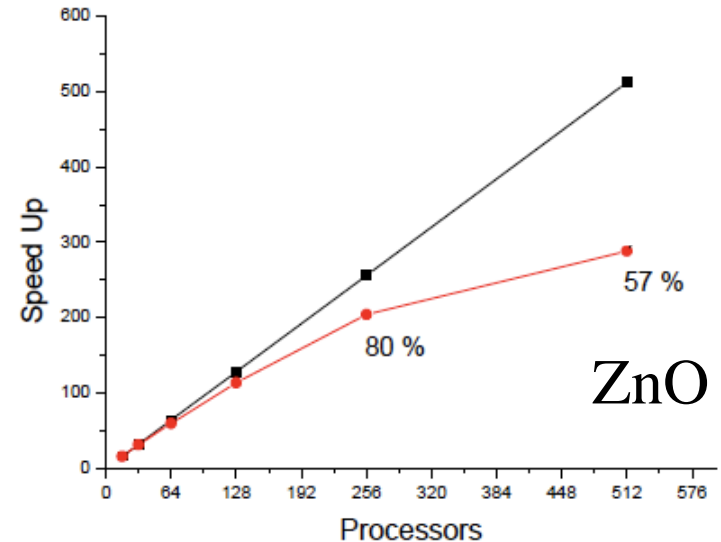


FIG. 1. Difference between DFT and QSGW calculations for model III of (a) the planar average of the electronic density and (b) the macroscopic average of the local potential. The density is expressed in  $m_e/a.u.$ , and the potential in  $meV$ .

## Si/SiO<sub>2</sub> interface

[Shaltaf, Rignanes, XG, Giustino, Pasquarello, Phys. Rev. Lett. 100, 186401 (2008)]



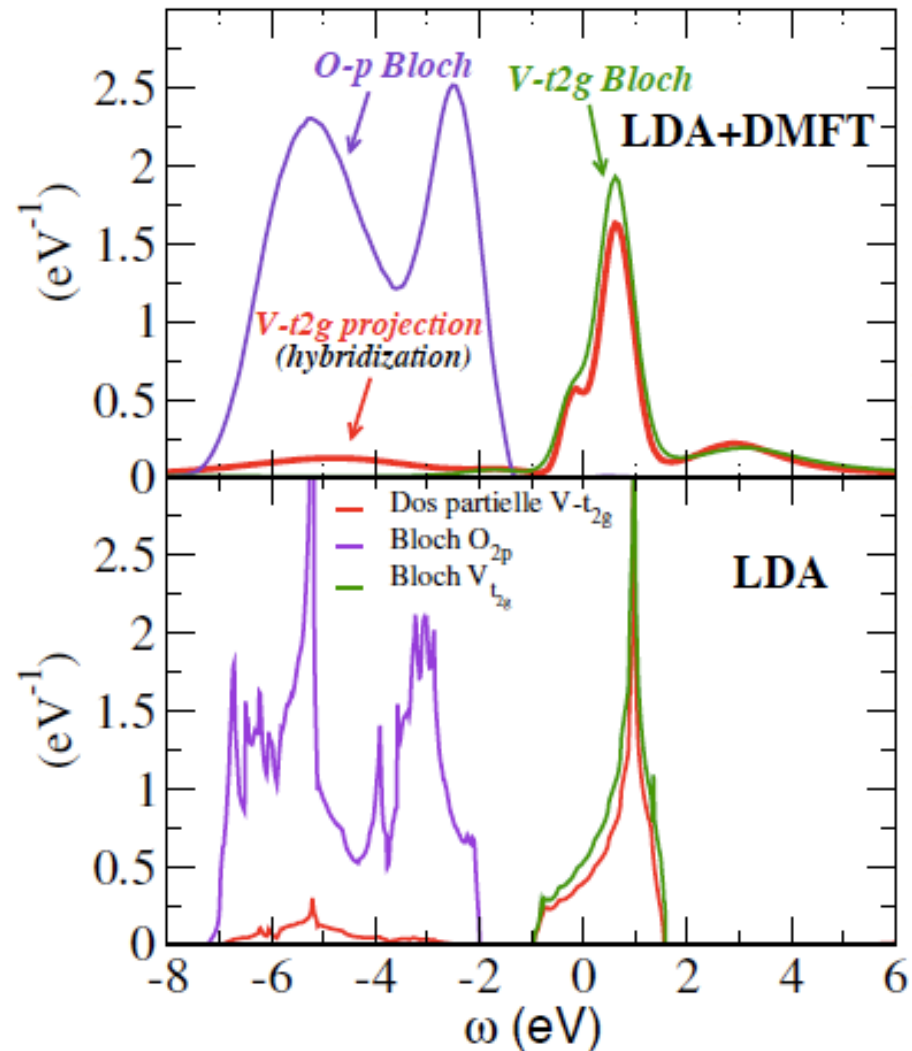
Extrapolar technique [F. Bruneval & XG, Phys. Rev. B 78, 085125 (2008)]  
decrease by a factor of 4...8 the number of conduction states needed.

Calculations done up to 200 atoms. Routine for up to 100 atoms.

# DFT + Dynamical Mean Field Theory

(Amadon, Lechermann, Georges, Jollet, Wehling, Phys. Rev. B. 77, 205112 (2008))

DOS SrVO<sub>3</sub>



# Wavelets + hybrid GPU/CPU (BigDFT)

(Genovese et al, J. Chem. Phys 131, 034103 (2009))

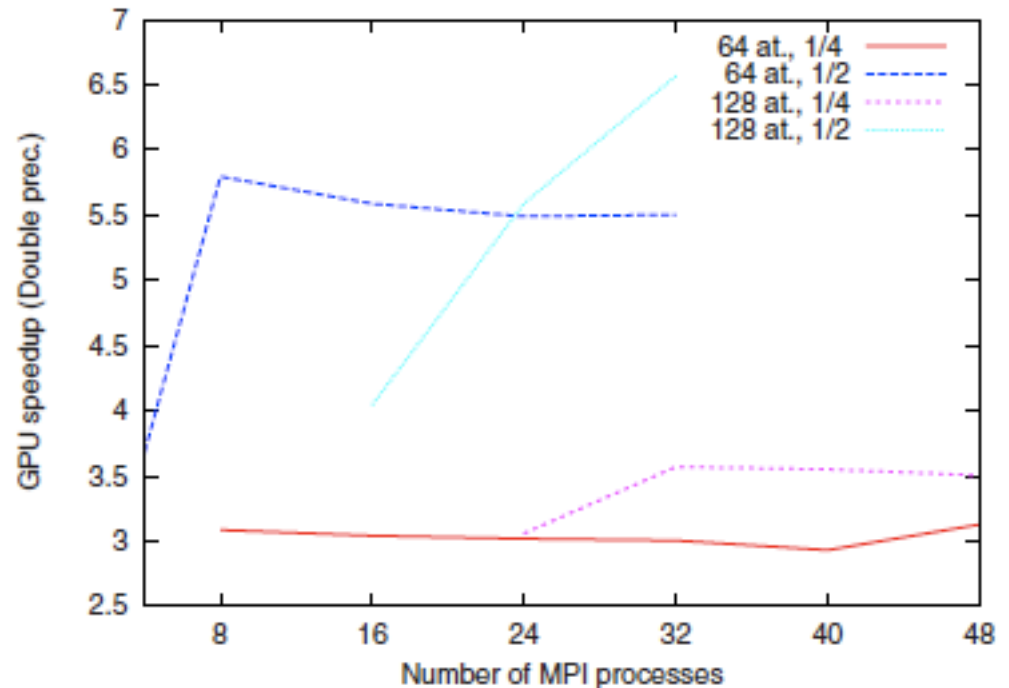
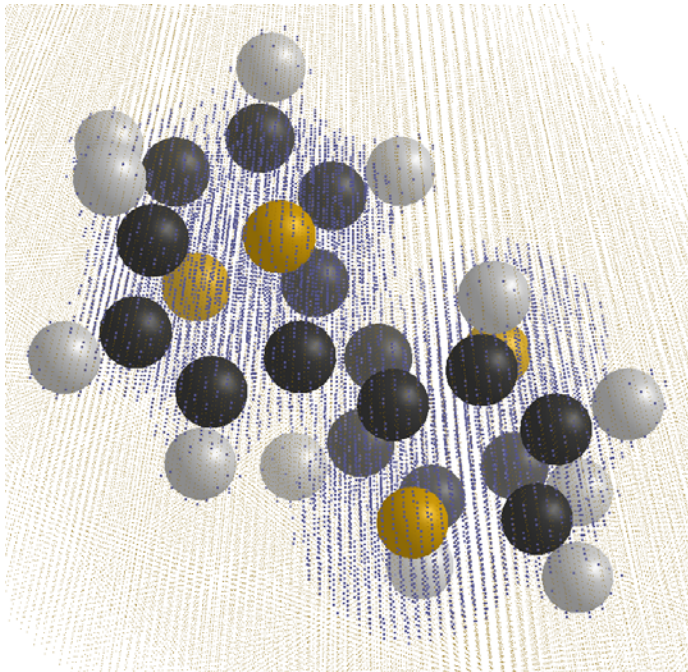


FIG. 5. Speedup of the full DFT code as a function of the number of CPU cores (i.e., MPI processes), when the number of CPU cores associated with the same GPU is two (1/2 curves) or four (1/4 curves). Systems with 64 and 128 atoms are analyzed, showing no significant differences in the behaviors.

# The broader picture ... couplings ...

- Through files :
  - With pseudopotential generators  
(FHI98PP, OPIUM, APE, ATOMPAW, USPP, UPF ...)
  - With file “postprocessors”  
(EXC, DP, YAMBO, V\_SIM, XCRYSDEN ...)
- Through libraries :
  - LibXC (from OCTOPUS), WANNIER90,  
macroave (from SIESTA), vdW-DF (Soler,Langreth)
- Also with packagers :
  - Debian, Gentoo, Ubuntu



fhi98pp



Home



European Theoretical Spectroscopy Facility



**Intranet**

Welcome to the European Theoretical Spectroscopy Facility



# LibXC (M. Marques – from OCTOPUS)

- Library of exchange correlation functionals
- 19 LDA
- 55 GGA
- 24 Hybrids
- 7 Meta-GGA
- Need appropriate input depending on the kind of functional  
(density, gradient of density, laplacian of density, kinetic energy density)  
LDA, GGA, meta-GGA interfaced with ABINITv6

# ABINIT + YAMBO

Exciton-plasmon states in nanoscale materials :  
breakdown of the Tamm-Dancoff approximation.

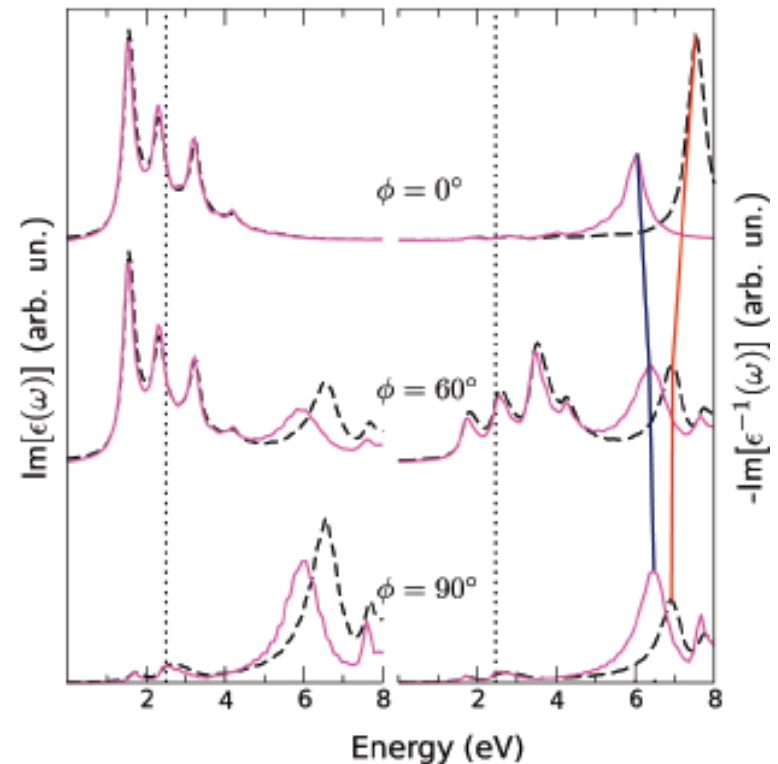
M. Grüning, A. Marini & XG, Nanoletters 9, 2820 (2009)

- Understanding that the Bethe-Salpeter Hamiltonian is pseudo-Hermitian
- New Lanczos type algorithm to deal with pseudo-Hermitian Hamiltonians
- This allows an easy computation of exciton-plasmon coupling for reasonably large systems, here CNT zig-zag (8,0).

Red : full Hamiltonian

Dashed line : Tamm-Dancoff approx.

**Figure 3.** Polarized (angle  $\phi$  with respect to the CNT axis) absorption (left stack) and  $\phi$ -dependent EEL (right stack) spectra of zigzag (8,0) CNT calculated within the BS equation either by using the full (solid line) or the TDA (dashed line) Hamiltonian. The dotted line indicates the position of the quasiparticle band gap.<sup>6</sup> In the EEL spectra (right stack), the blue and red lines highlight the angular dependence of the main peak at  $\sim 6$  eV for the full Hamiltonian and TDA respectively. Going from  $\phi = 0^\circ$  to  $\phi = 90^\circ$  the peak is red shifted by about 1 eV within the TDA while is blue shifted by about 0.5 eV in the full Hamiltonian. See also Figure 2.



# **Software engineering in ABINIT**

# Software engineering concepts

Our expertise ... is NOT software engineering !

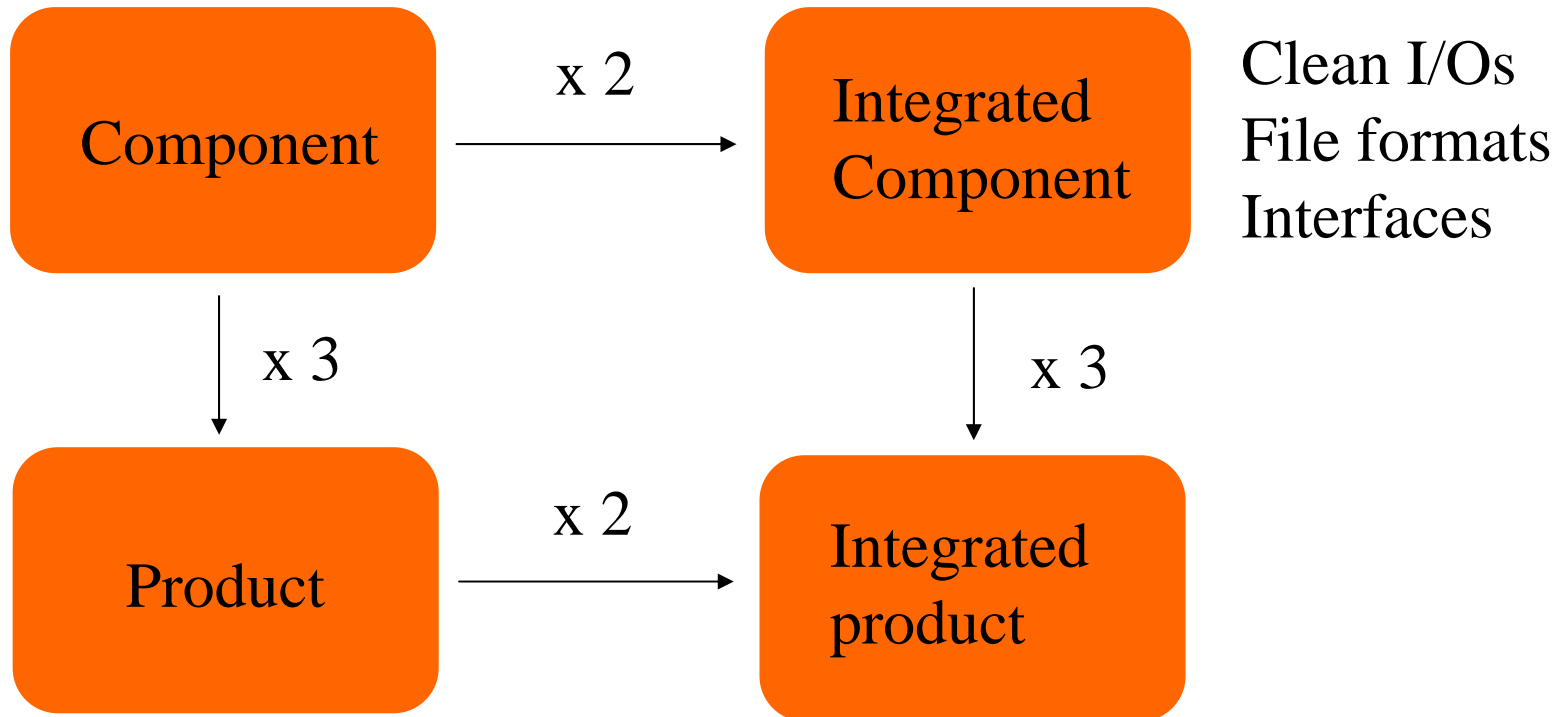
What is software engineering ?

- Not merely the fact of switching from FORTRAN to C++...
- A **human** science : How to improve the **developer**'s productivity ?
- Potentially very important to us ...

Questions :

- ▣ What really takes time ?
- ▣ Can (software) tools improve the productivity ?
- ▣ How to make a group have better productivity ?

# What takes time ?



Clean I/Os  
File formats  
Interfaces

Documentation  
Testing, portability  
Maintenance

This is what we want to  
rely on, for our group  
long-term development !

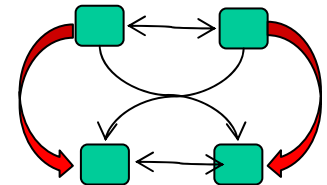
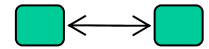
# The essence of software programming

*“a scaling-up of a software entity is not merely a repetition of the same elements in larger size; it is necessarily an increase in the number of different elements. In most cases, the elements interact with each other in some nonlinear fashion, and the complexity of the whole increases much more than linearly”*

*(F. Brooks, The mythical man-month. Essays on software engineering, 1995 20th Anniversary edition)*

- Brooks argued that essence is a substantial part of the development time, that it is inherently complex and ... **incompressible !**

(the bearing of a child takes nine months, no matter how many women are assigned)



# Software reuse

How to address the “essence” bottleneck ?

- Use already existing software ! **Software re-use.**
  - ▣ Conceptual work already done
  - ▣ Debugged, tested, I/O set-up !
  - ▣ Add “integrated product” to the system (**need adequate licence**)
  - ▣ Can be completely external
  - ▣ Can be internal re-use : need modularity !
  - ▣ Grow, not build software : incremental development. Also psychological: one has something that works.

**Building software takes some uncompressible human time, even if we eliminate the accidental difficulties, and attack the essential difficulties in an efficient way.**

# Maintenance

- Program maintenance : unlike for a car, no cleaning, lubricating, repair of deterioration. The needed changes repair **design “defects”**. These appear because of new functionalities to be implemented. Other modifications are due to **change of environment** (hardware, compilers)
- Moreover, fixing a defect has a substantial (20%-50%) chance of introducing another ! Importance of **automatic testing**.
- The only constancy is change itself. So, plan the system for change... And have tools for **version maintenance** ...

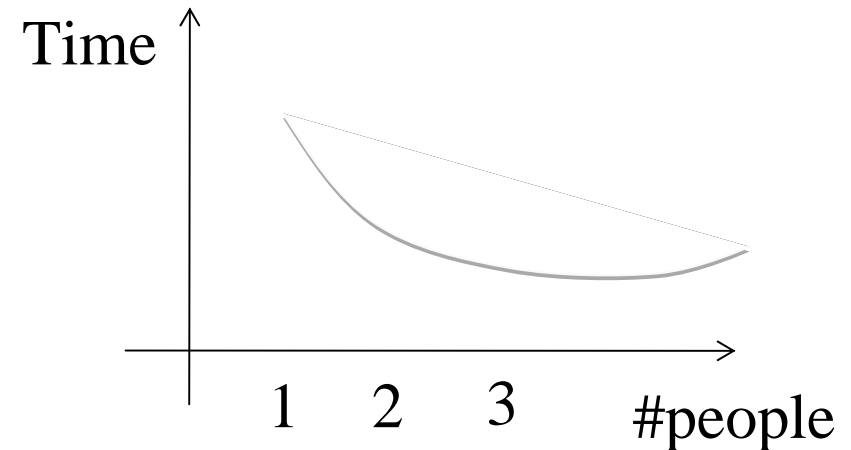


# Group effort

How to make a **group** have a better productivity ?

- First, each person need training
- Then, software construction is a **system** effort

⇒ communication effort  
can dominate the decrease  
in individual task time  
brought by partitioning.



Need **asynchronous** work, in view of building  
**integrated** products.

Doing top-level work before delivery pays off :  
less communication needed (complaints, fixes ...);  
but training to do top-level works takes time !

# ABINIT Distributed development

How to secure existing capabilities despite the development efforts (by rather diverse groups) and the associated bug generation ?

*More than 500 automatic tests have been set up, and new ones are added for securing each new feature.*

Different groups use different platforms ... and how to reuse software

*The build system relies on the “autotools” (itself software reuse).*

*The build system has been designed for the easy inclusion of libraries (or “plug-ins”)*

Different groups might have different coding style ...

*The (F90) routines follow explicitly stated coding rules (ABINIT style), with a special format that allows processing by ROBODOC, and different utilities (in perl and python) that allow global management of sources.*

How to synchronize the development effort ?

*Use of a central repository, worldwide accessible, using the GNU Arch model, as implemented in the software BAZAAR.*

**Implementation of the “self-testing” and “self-documentation” software concepts, as well as use of recent software engineering management tools.**

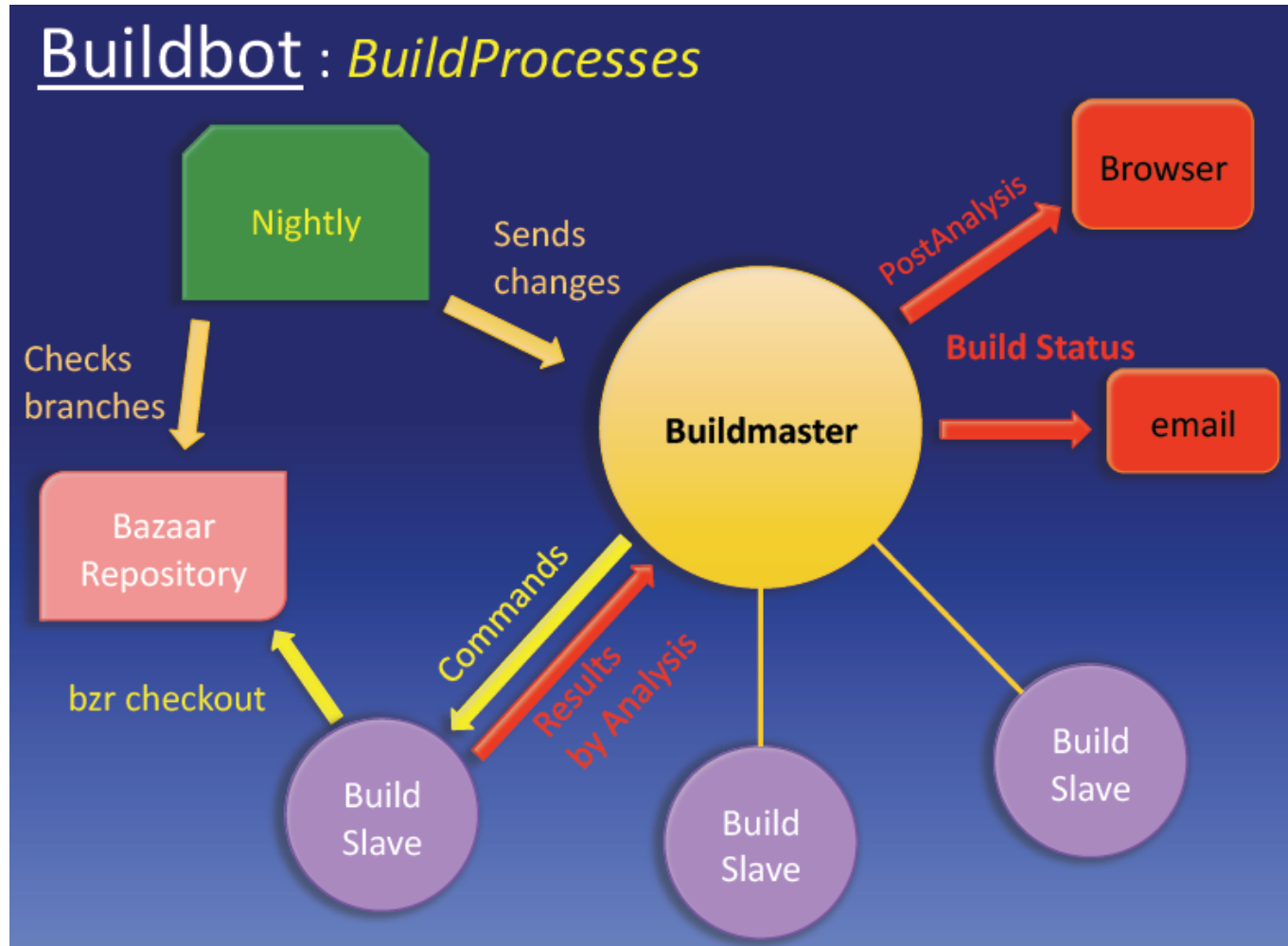
# Version control : bazaar (aka bZR)

In 2007 we (finally) reached a first-class version control system with **bZR** ...

- Very easy to use and install
- Each developer has one or more branches, that are easily merged with other branches (powerful merge algorithm ; of course, there might be conflicts to be resolved at merge time ...)
- Worldwide access of the repository (everybody can see everybody's work)
- Distributed version control system : easy to work with no access to the net, then synchronize with the repository when the net is accessed.
- Renaming files (or directories) does not break history

# ETSF Test farm, under control of buildbot

- At present : 10 slaves, each with 4 cores
- 8 different platforms (compiler/architecture)



# Automatic analysis of tests

```

pouillon_5.9.3-public/r711
=====
Tests SEQ start at 22:36 and done after      1433s

test built_in OK

=====
      Serie  #tests  #succes  #passed  #failed  #missing
=====
      bigdft   13     13       0       0       0
      etsf_io   8      8       0       0       0
      fast    27     27       0       0       0
      gwdp    32     32       0       0       0
      libxc    5      5       0       0       0
      tutorespfn 40     40       0       0       0
      tutorial 53     53       0       0       0
      v1      96     96       0       0       0
      v2      96     96       0       0       0
      v3      92     92       0       0       0
      v4     100    100       0       0       0
      v5     100    100       0       0       0
      wannier90 3      3       0       0       0
      paral   74     74       0       0       0
=====

Paral Tests DONE ( time elapsed: 635s )
=====
Powered by Analysis V2.2.1

```



# Automatic on-line results

<b>ABINIT</b> last build		build successful	build successful	build successful
current activity		idle	idle	idle
time (CEST)	changes	<u>testf_gcc44</u>	<u>testf_gcc44_serial</u>	<u>green_intel10</u>
<b>14:35:15</b>	<u>trunk</u>		cleaning done <u>stdio</u>	
		cleaning done <u>stdio</u>	succeeded <u>stdio</u>	
		failed (98) <u>stdio</u>	uploading summary.log	succeeded <u>stdio</u>
<b>14:23:44</b>		uploading summary.log	seq tests done <u>stdio</u>	uploading summary.log <u>stderr</u>
		all tests done <u>stdio</u> <u>xreport</u> <u>extralog</u> <u>full_output</u> <u>fidifflog</u> <u>summary</u>	<u>xreport</u> <u>extralog</u> <u>full_output</u> <u>fidifflog</u> <u>summary</u>	checkout done failed
		downloading to Analysis	downloading to Analysis	cleaning done <u>stdio</u>
<b>13:49:32</b>		make mj4 done <u>stdio</u>	make mj4 done <u>stdio</u> make	set props: username compilo version

# Restructuring the package

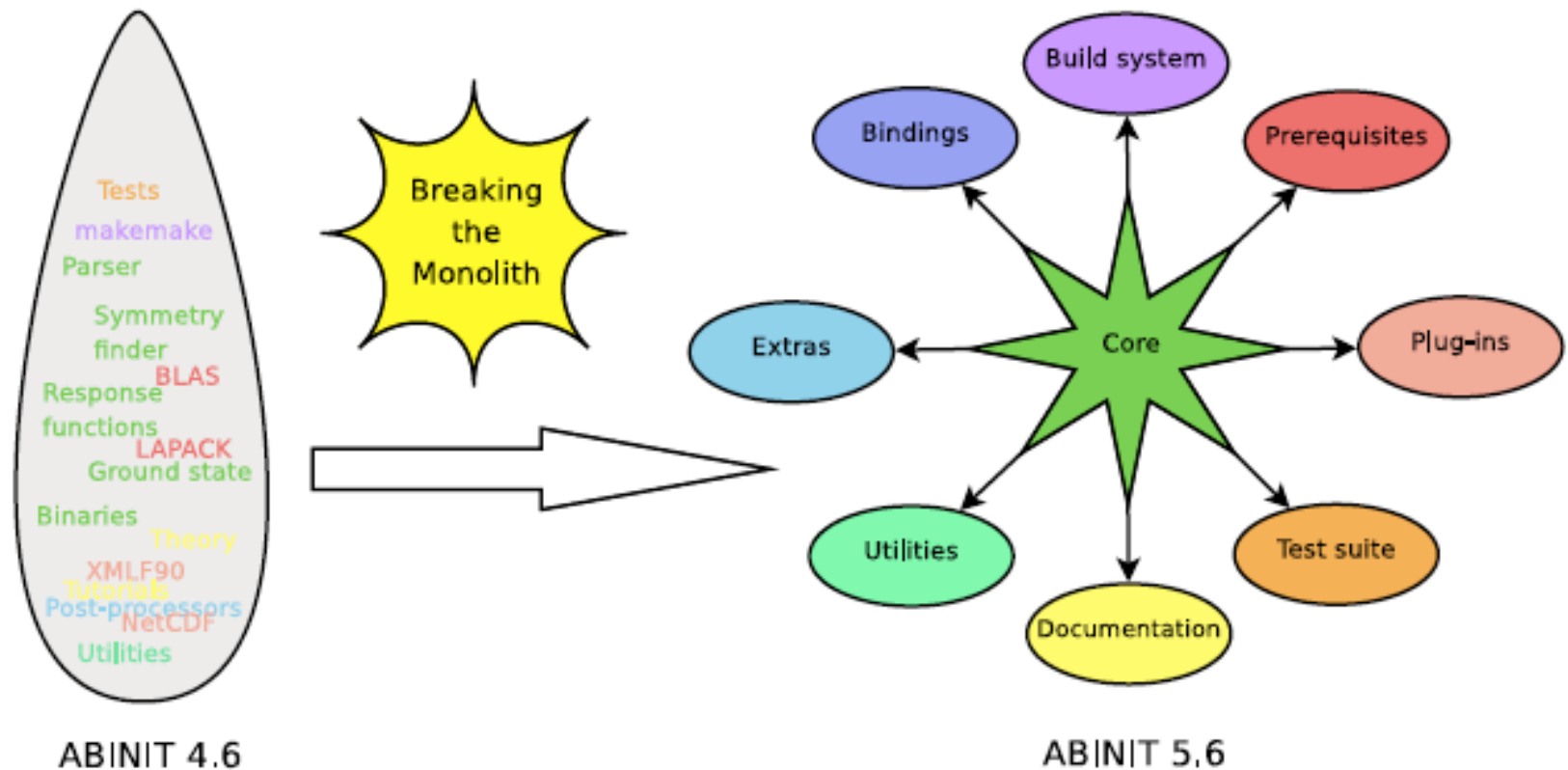


Figure 1: Overall view on the mutation of ABINIT, based on the restructuring of the code and the implementation of a new build system, providing better modularity and extensibility.

Compliance with packaging standards

Integration in Debian, Ubuntu, and Gentoo packages

See Y. Pouillon & XG, psi-k newsletter 90 highlight of the month (Dec. 2008)

# Software Reuse

## ○ Re-use : external

- ▣ F90 : MPI/OpenMP/MPI\_IO, BLAS/LAPACK/SCALAPACK, CPP, NetCDF, ETSF\_IO, LibXC, Wannier90, CUDA
- ▣ Environment : bzd, ROBODOC, AUTOTOOLS, PYTHON scripts, PLONE, Buildbot, loggerhead, mailing list system

## ○ Re-use : internal

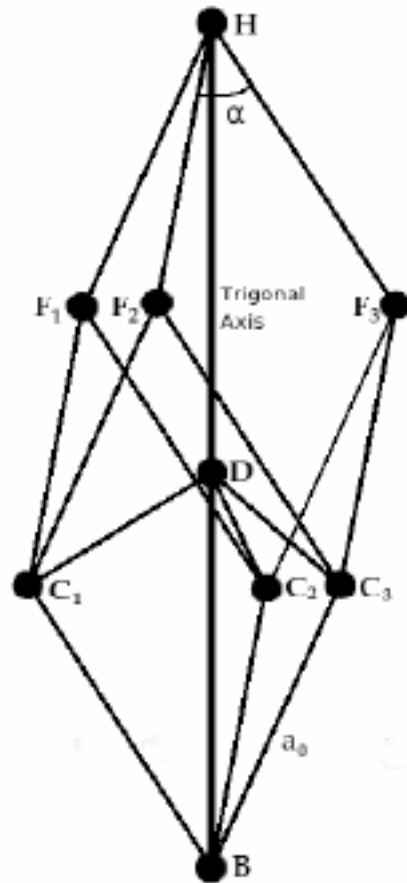
- ▣ reuse needs modularisation
- ▣ reuse needs documentation
- ▣ reuse needs the appropriate build system ...



# **Spin-orbit in density-functional perturbation theory**



# Bismuth : cell geometry



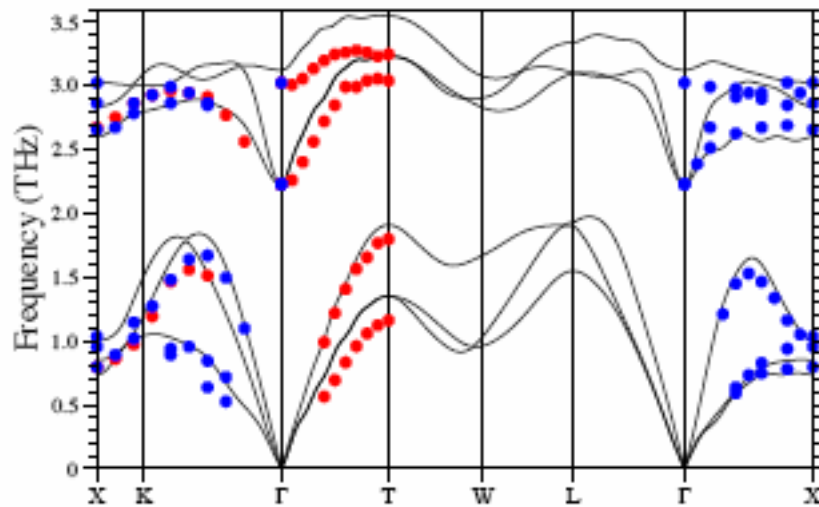
Rhombohedral A7 unit cell  
2 atoms per unit cell

TABLE I: Crystallographic parameters for the unit cell of Bi.

	Non-metallic without SO	Metallic without SO	with SO	Experiment values
$a_0$ (Å)	4.6496	4.6525	4.6944	4.7236
$\alpha$ (deg)	57.57	57.48	57.57	57.35
$z$	0.47102	0.47108	0.46819	0.46814
$a_{NN}$ (Å)	3.0385	3.0383	3.0505	3.0624
$\overline{BD}$ (Å)	5.4610	5.4684	5.4805	5.52252
$\overline{DH}$ (Å)	6.1328	6.1396	6.2250	6.27421

1% change of lattice parameter  
z-parameter perfect with SO

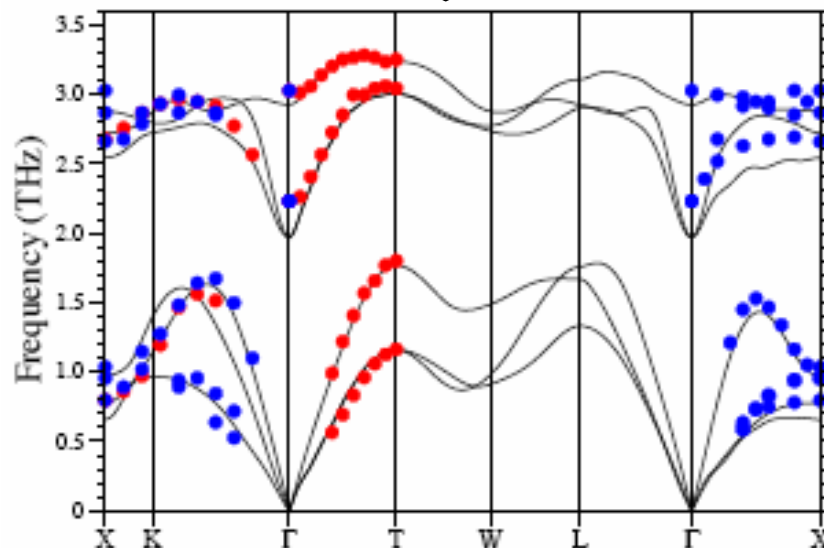
# Bismuth : phonon band structure



Full line DFPT **without** spin-orbit

Only very old experimental data available

- Yarnell et al, IBM J.Res. Dev. 1964
- Smith, internal report Los Alamos 1967



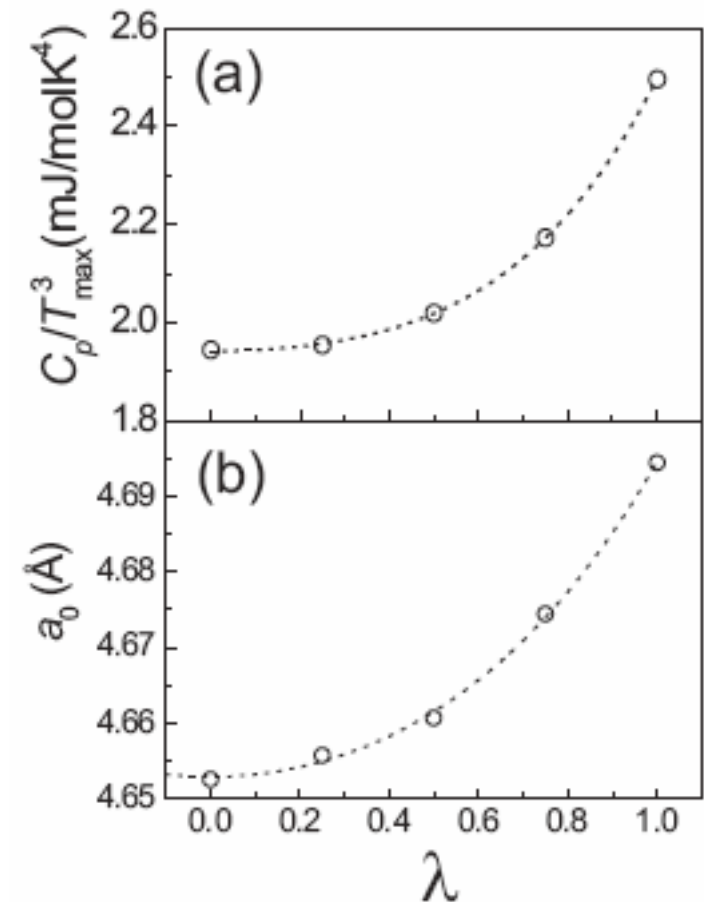
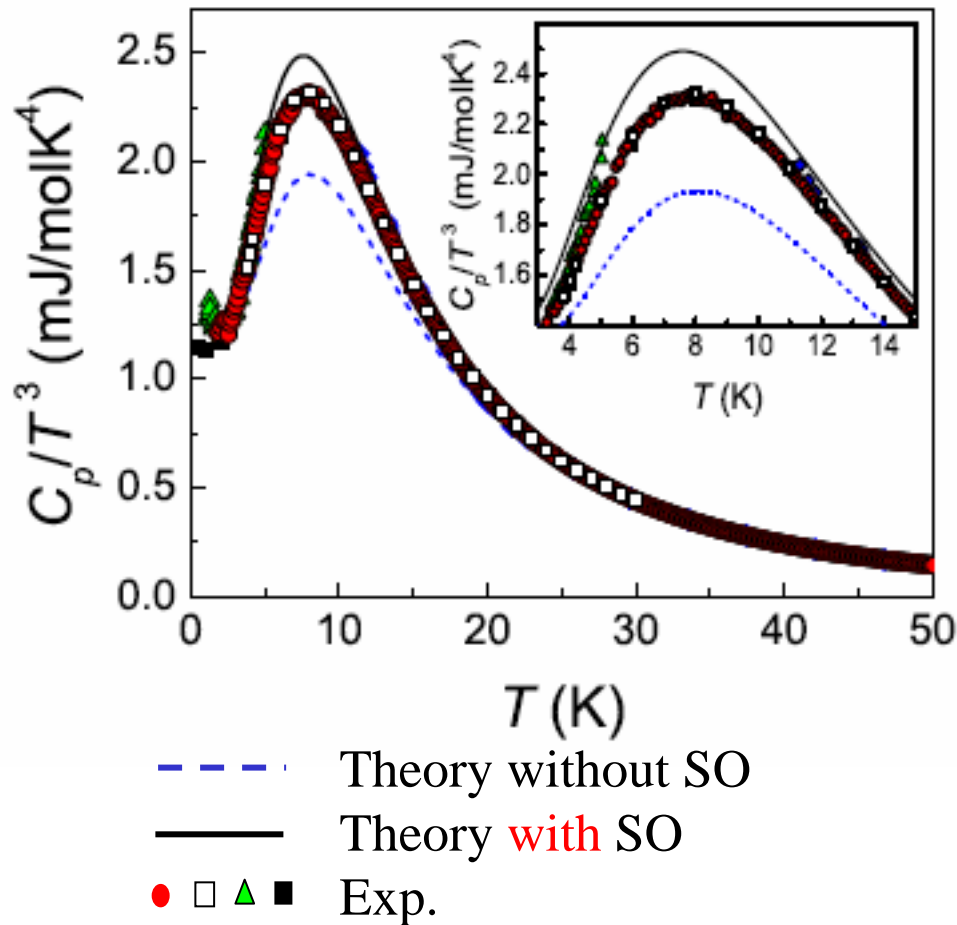
**10-15% change due to SO**

Full line DFPT **with** spin-orbit

LE. Diaz-Sanchez, A.H. Romero, XG, Phys. Rev. B 76, 104302 (2007)

# Bismuth : specific heat

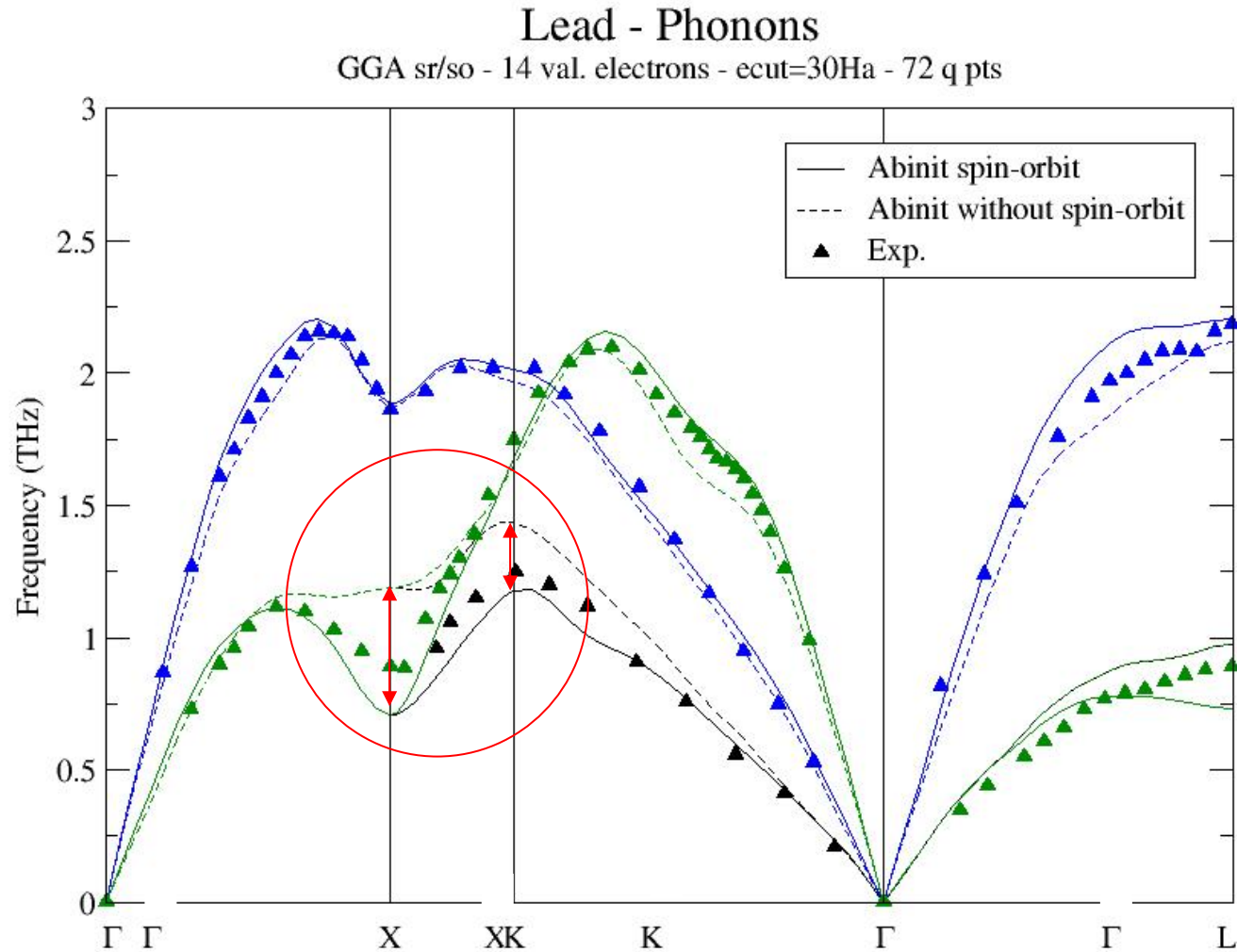
(based on Bose-Einstein statistics for phonons)



LE Diaz-Sanchez, AH Romero, M Cardona, RK Kremer, XG, PRL 99, 165504 (2007)

# Lead

Th/Exp discrepancy recognized in 1996 (Liu & Quong PRB53, R7575)



M. Verstraete, M. Torrent, F. Jollet, G. Zerah, X.G, PRB 78, 045119 (2008)

# Summary

Need for group development

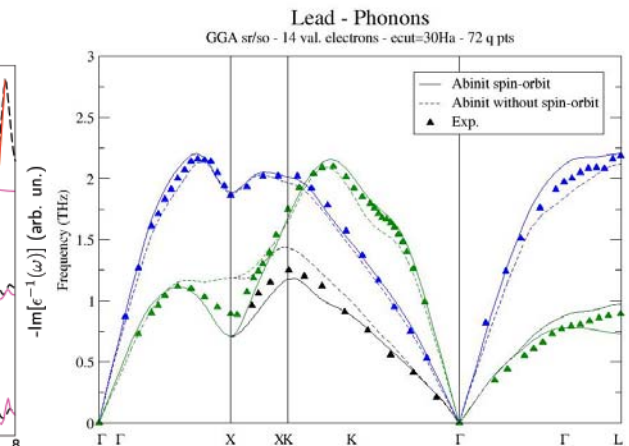
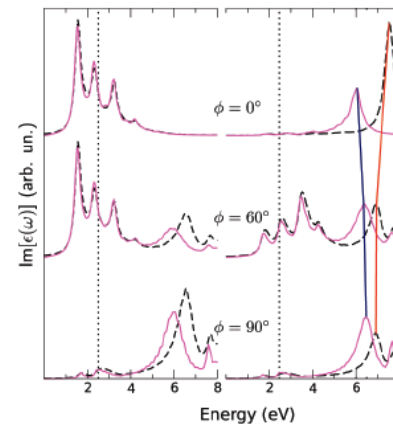
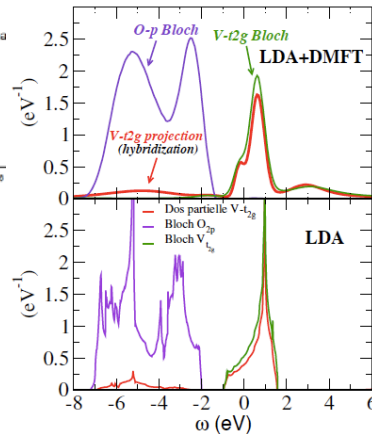
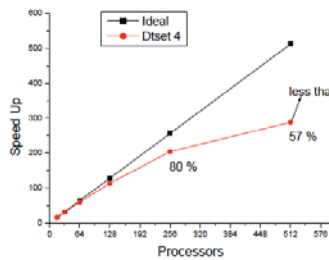
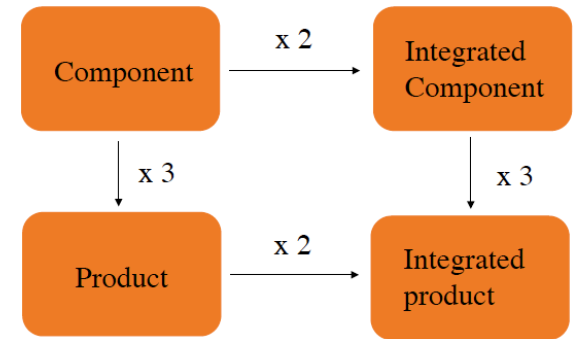
Need for interaction with other communities

Importance of software engineering concepts,  
especially for group development

Incompressible time => **Software re-use**

Tools and software techniques can help  
(version control system, build system)

ABINIT : integrating many capabilities



# Thanks to you and ...

UCL : J.-M. Beuken, A. Jacques, G.-M. Rignanese, S. Leroux, M. Giantomassi, F. Da Pieve,  
A. Lherbier, T. Rangel, M. Oliveira

CEA-Bruyeres (France) : M. Torrent, F. Jollet, G. Zérah, B. Amadon, G. Jomard, M. Mancini,  
F. Bottin, S. Mazevet, M. Delaveau

U.Liège (Belgium) : M. Verstraete, Ph. Ghosez, J.-Y. Raty, P. Hermet

CEA-Grenoble (France) : T. Deutsch, D. Caliste

U. San Sebastian (Spain) : Y. Pouillon

CEA-Gif-sur-Yvette (France) : F. Bruneval

Ecole Polytechnique Palaiseau (France) : L. Reining

Rutgers U. : D. Hamann

U. Milano (Italy) : G. Onida, D. Sangalli

U. Basel (Switzerland) : S. Goedecker

ESRF Grenoble (France) : L. Genovese

Dalhousie U. (Canada) : J. Zwanziger

U. Caen (France) : P.-M. Anglade

U. Amman (Jordania) : R. Shaltaf

U. Montréal (Canada) : M. Coté, P. Boulanger

ENS Lyon (France) : R. Caracas

U. Grenoble (France) : V. Olevano

Corning Inc. (USA) : D.C. Allan

Mitsubishi Chemical Corp. (Japan) : M. Mikami

U. York : R. Godby

