Computing at CERN - II

Summer Student Lectures 2002

Jamie Shiers

http://cern.ch/jamie/

Lecture II

- Computing at CERN Today

➢ **Software at CERN Today**

- The future & LHC Computing

Introduction

- For a long time it puzzled me how some **g edge**, could
- and t a **comp** n the abilit ngs, ...
- while smart peopl edibly stupi
- They **tch**.

Bill Bryson: "Notes from a Big Country".

Homework

Review of homework from lecture I

Exercise I

- Implement a Unix utility (grep, cron, …) according to *man* specification
- You don’t actually need to do the exercise – just pretend you have!

Software

Producing high-quality software is:

- Far from easy
- Far from cheap
- Still not a solved problem

Anyone can program?

- “Everyone can be taught to sculpt: Michelangelo would have had to be taught not to. So it is with great programmers.”

Overview

- Software Engineering
- Software Process
- Real examples from CERN

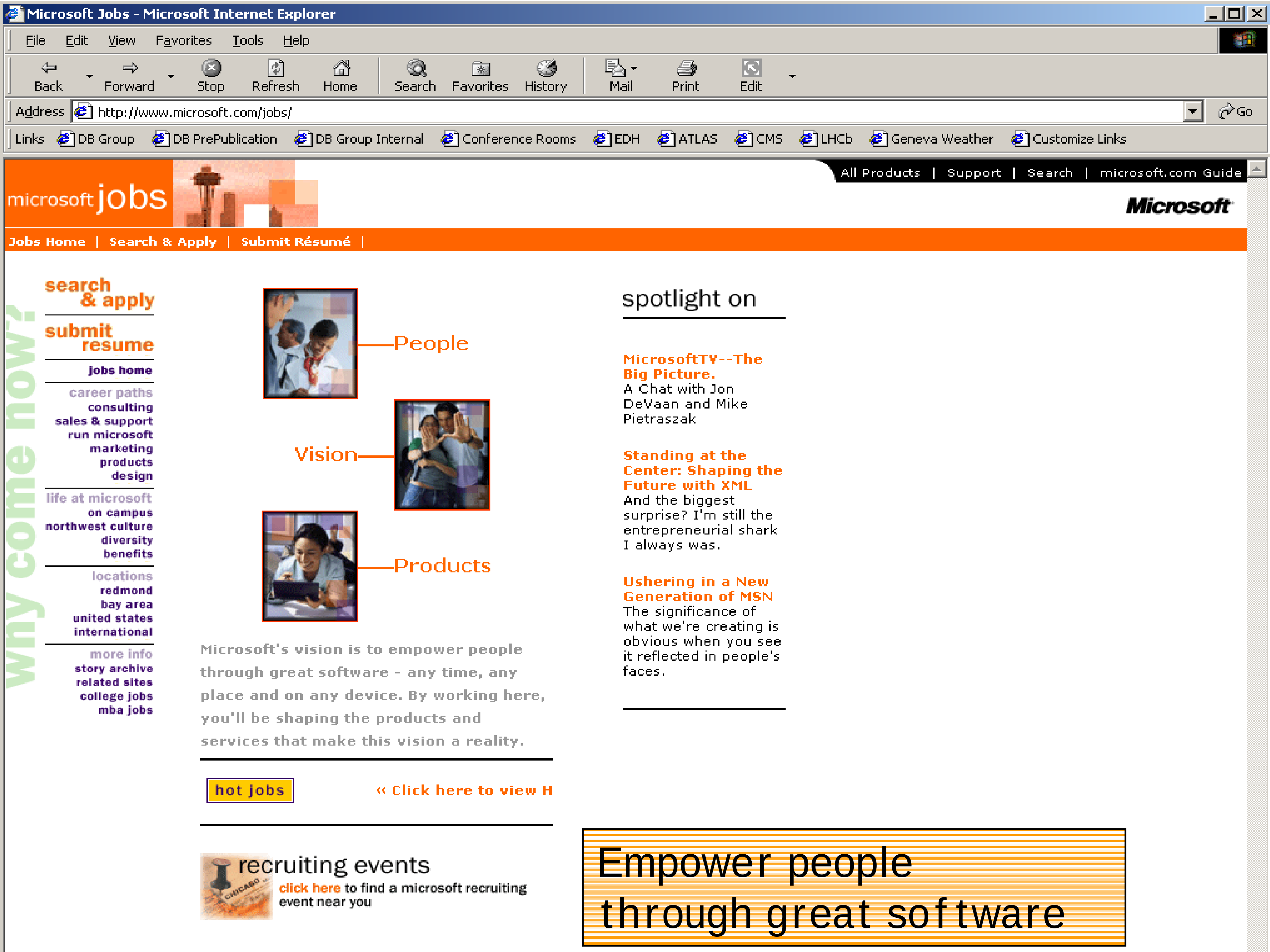
Disclaimer

- CERN and its collaborators have produced a vast quantity of high-quality, well documented software
- Well disciplined approaches are in use in many areas of CERN
- Many people have devoted significant effort to improve the overall software process at CERN

Some Large Producers…

- Microsoft

- Oracle


Microsoft Jobs - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
Address http://www.microsoft.com/jobs/
Go
Links DB Group DB PrePublication DB Group Internal Conference Rooms EDH ATLAS CMS LHCb Geneva Weather Customize Links
microsoft jobs
All Products | Support | Search | microsoft.com Guide
Microsoft
Jobs Home | Search & Apply | Submit Résumé |
why come now?
People
Vision
Products
hot jobs
CHICAGO

search & apply

submit resume

jobs home

career paths
consulting
sales & support
run microsoft
marketing
products
design

life at microsoft
on campus
northwest culture
diversity
benefits

locations
redmond
bay area
united states
international

more info
story archive
related sites
college jobs
mba jobs

Microsoft's vision is to empower people through great software - any time, any place and on any device. By working here, you'll be shaping the products and services that make this vision a reality.

hot jobs « Click here to view H

recruiting events
click here to find a microsoft recruiting event near you

spotlight on

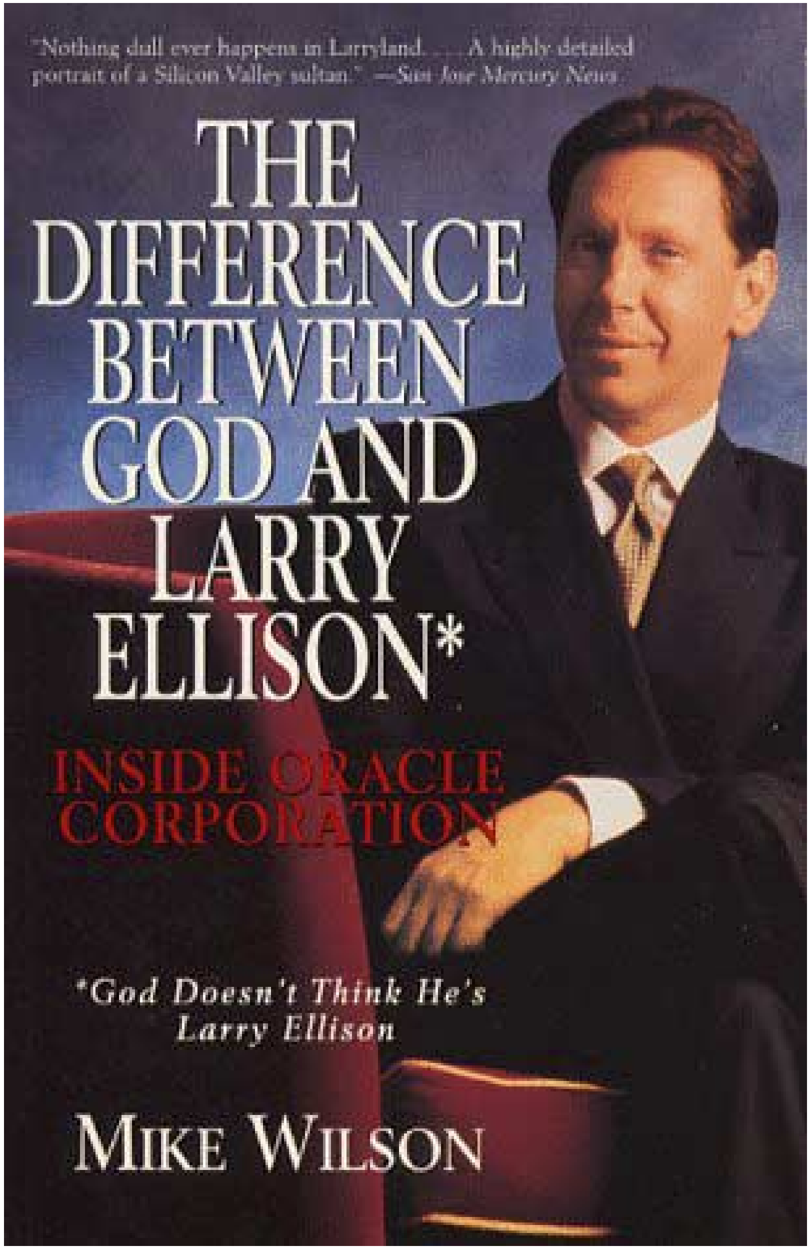
MicrosoftTV--The Big Picture.
A Chat with Jon DeVaan and Mike Pietraszak

Standing at the Center: Shaping the Future with XML
And the biggest surprise? I'm still the entrepreneurial shark I always was.

Ushering in a New Generation of MSN
The significance of what we're creating is obvious when you see it reflected in people's faces.

Empower people through great software

"I sense much NT in you!
NT leads to Blue Screen.
Blue Screen leads to downtime,
downtime leads to suffering...
NT is the path to the darkside!"

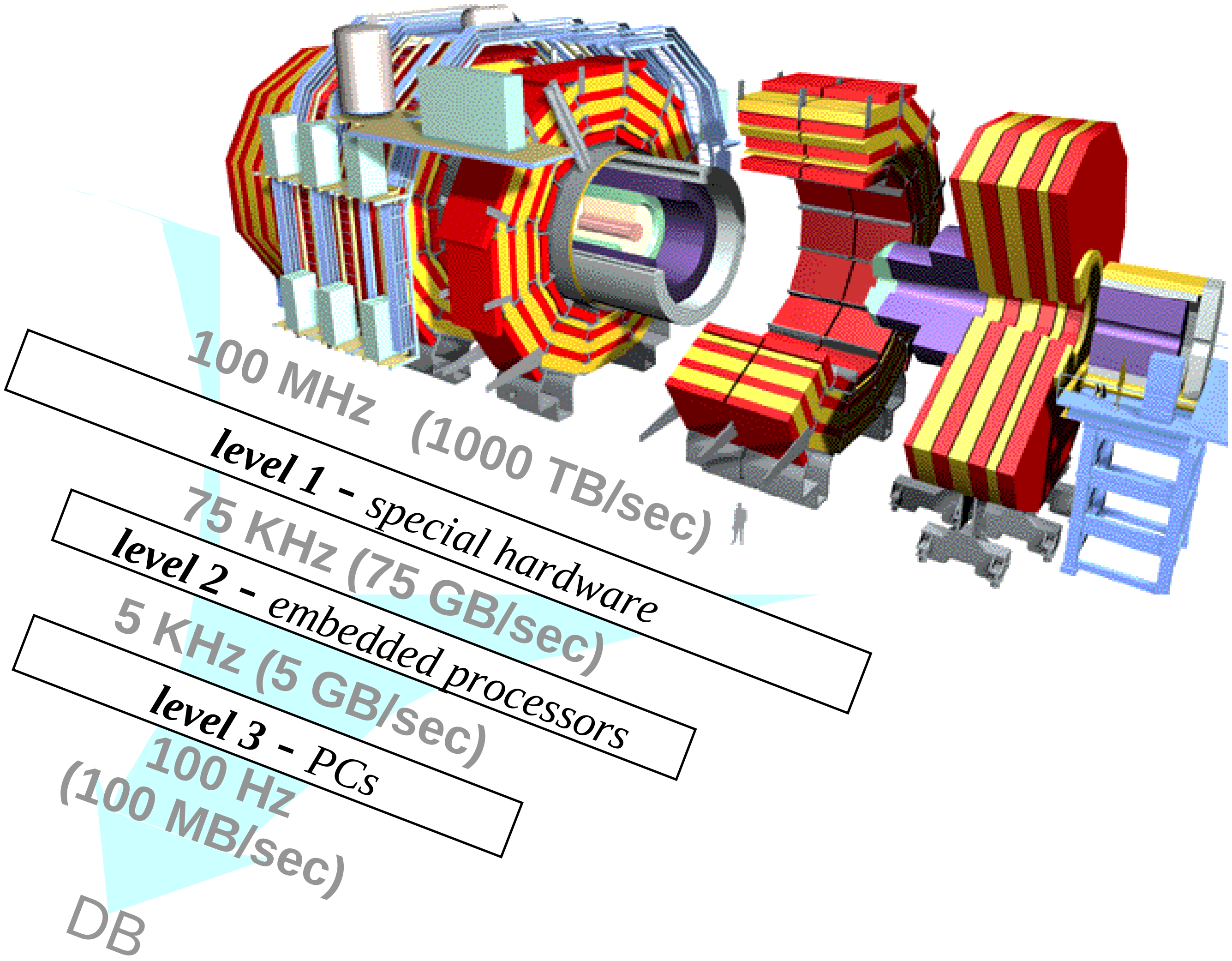

"Nothing dull ever happens in Larryland. . . . A highly detailed portrait of a Silicon Valley sultan." —San Jose Mercury News
THE DIFFERENCE BETWEEN GOD AND LARRY ELLISON*
INSIDE ORACLE CORPORATION
*God Doesn't Think He's Larry Ellison
MIKE WILSON

Why software quality?

- Airbus / BMW
- LHC data acquisition & processing

C M S

Compact Muon Solenoid



100 MHz (1000 TB/sec)

level 1 - special hardware

75 KHz (75 GB/sec)

level 2 - embedded processors

5 KHz (5 GB/sec)

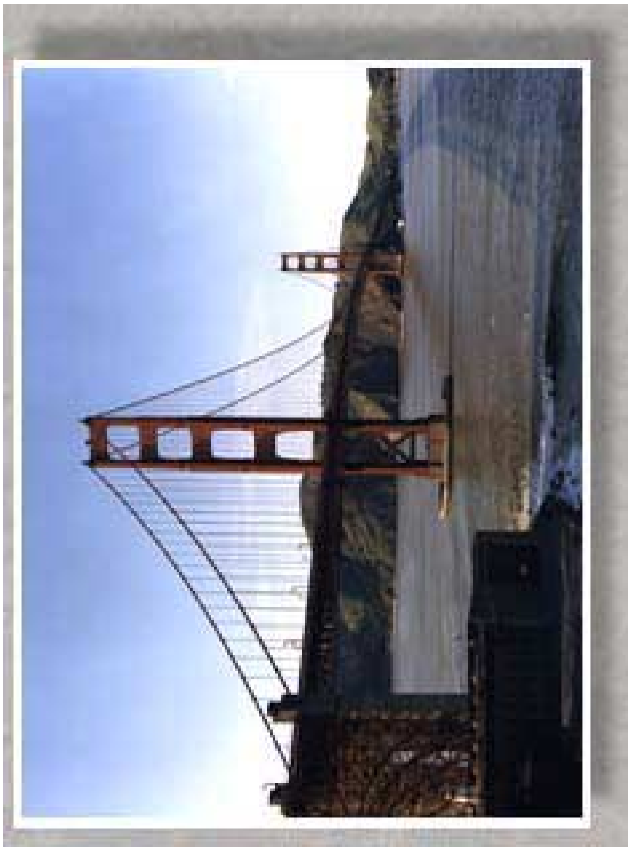
level 3 - PCs

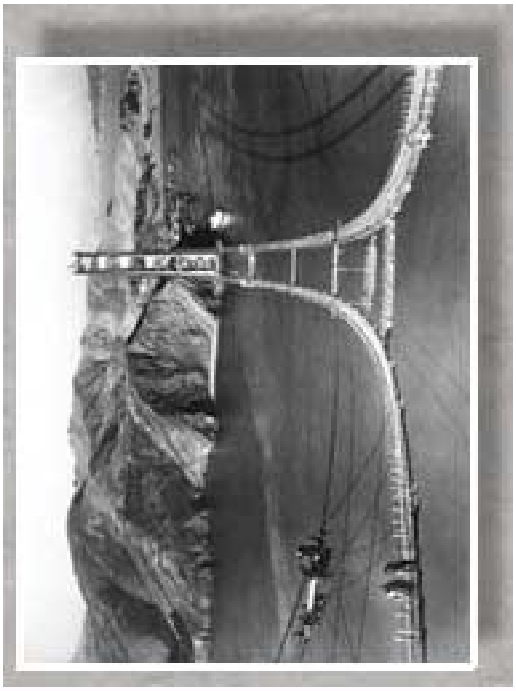
100 Hz (100 MB/sec)

DB

Software Engineering

- When discussing salary, its a profession;
- When discussing , Bugs, Errors and Liability, its a job;
- When discussing theory, its science;
- When discussing methods and practice, its engineering;
- When discussing the work and the work of others, its a craft;
- When managing it, its an art.





t

- So
- So
- So

[Jacquard-card Making.]

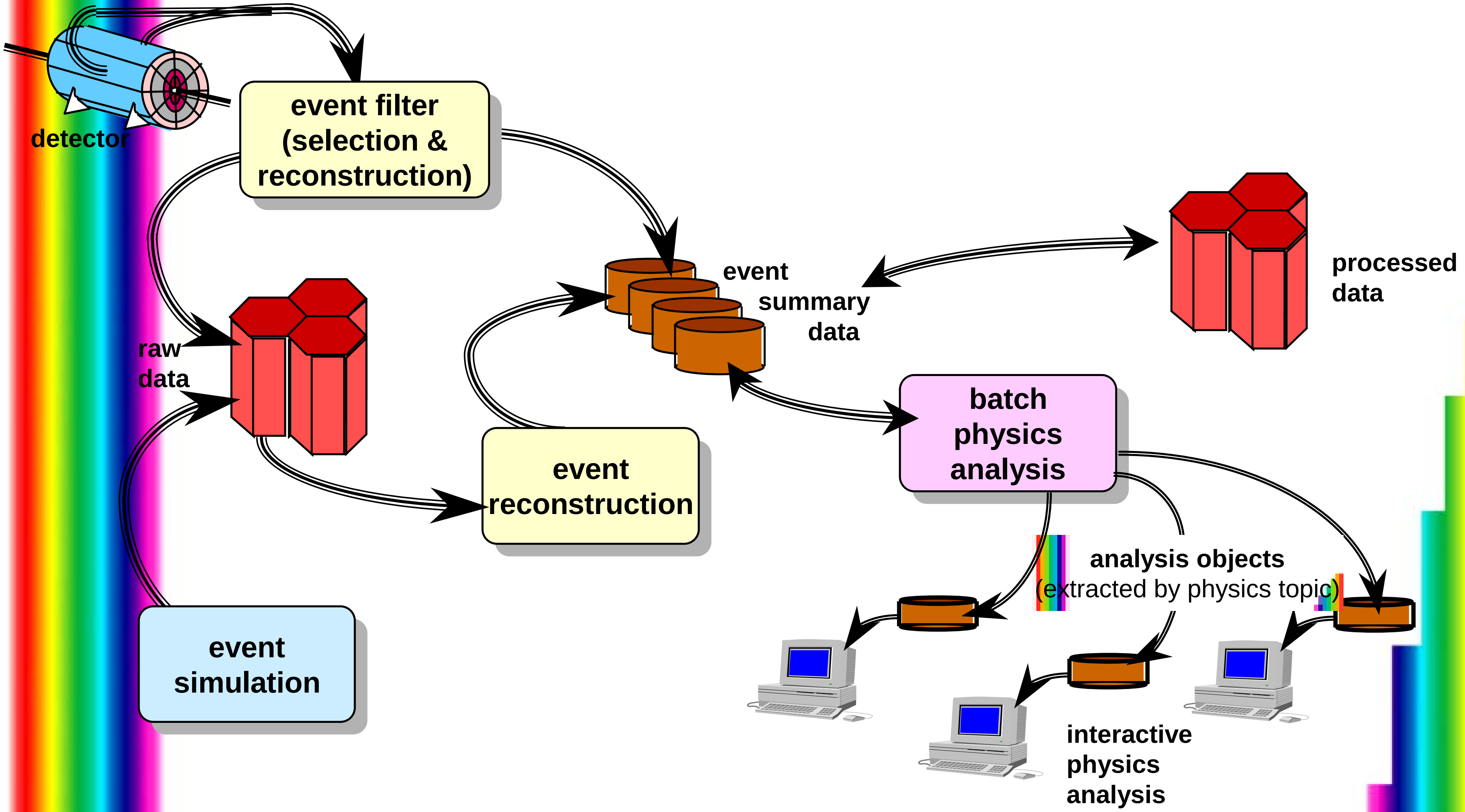
Software Engineering in HEP – The Reality

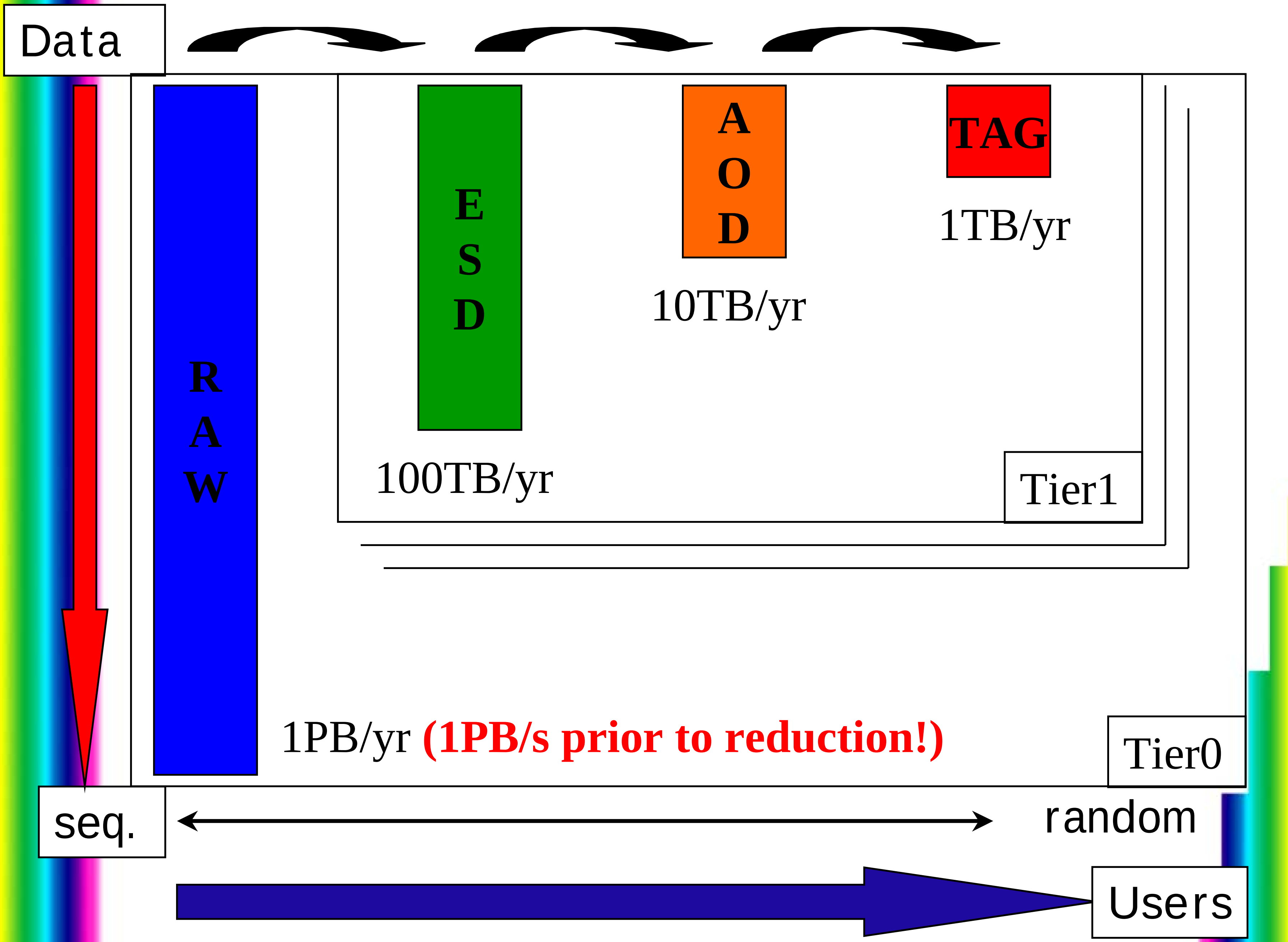
- Jürgen Knobloch, Computing in High Energy Physics, Tsukuba 1991
- **“In spite of all efforts, the most valuable tool is still a good Symbolic Debugger...”**

Software Complexity

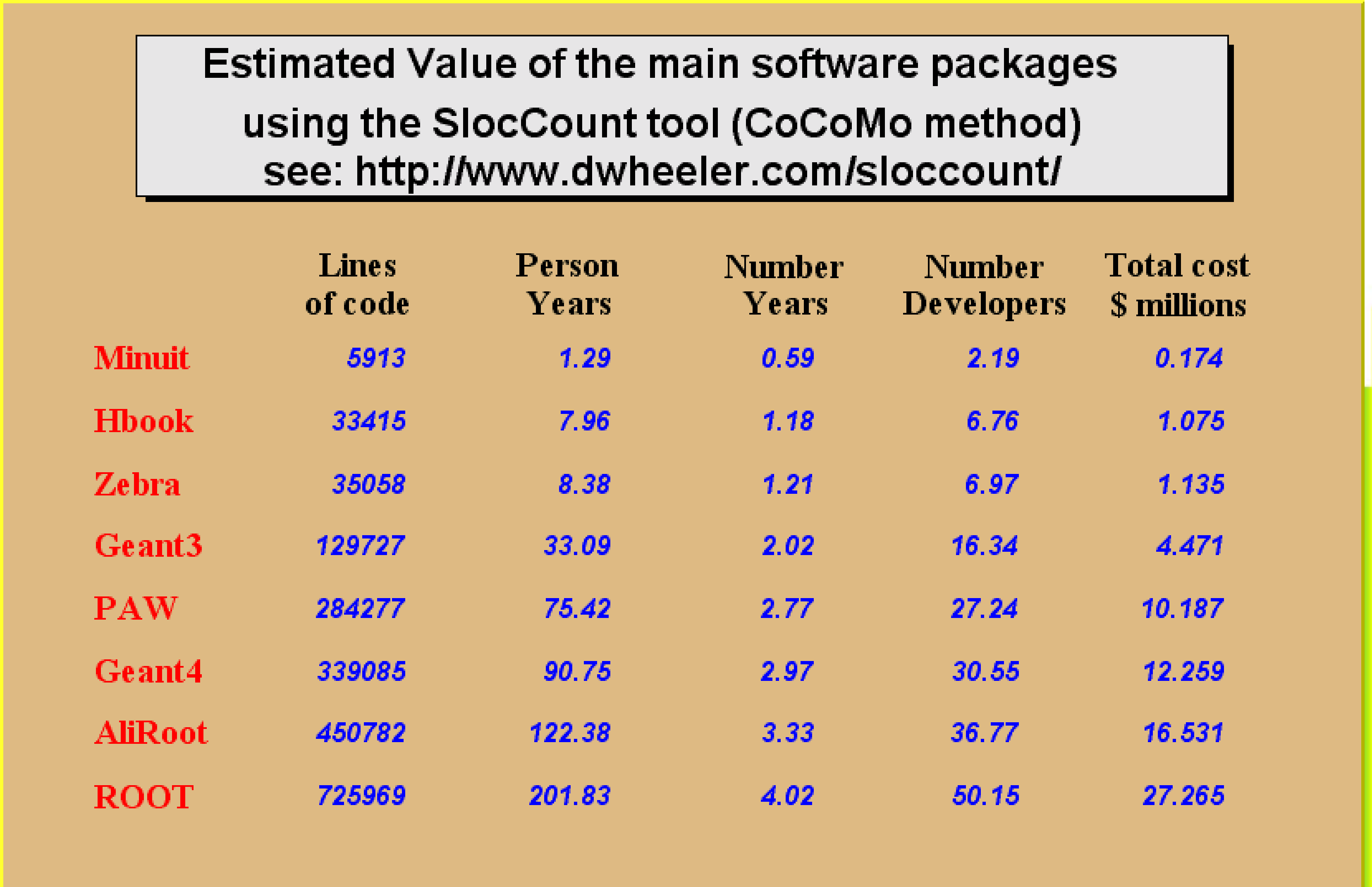
- Program complexity grows until it exceeds the capability of the programmer to maintain it.
- There are two ways of constructing a software design: one way is to make it so simple that there are obviously no deficiencies and the other way is to make it so complicated that there are no obvious deficiencies. The first method is far more difficult.

Data and Computation for Physics Analysis




Data
R A W
E S D
A O D
TAG
1TB/yr
10TB/yr
100TB/yr
Tier1
1PB/yr (1PB/s prior to reduction!)
Tier0
seq.
random
Users

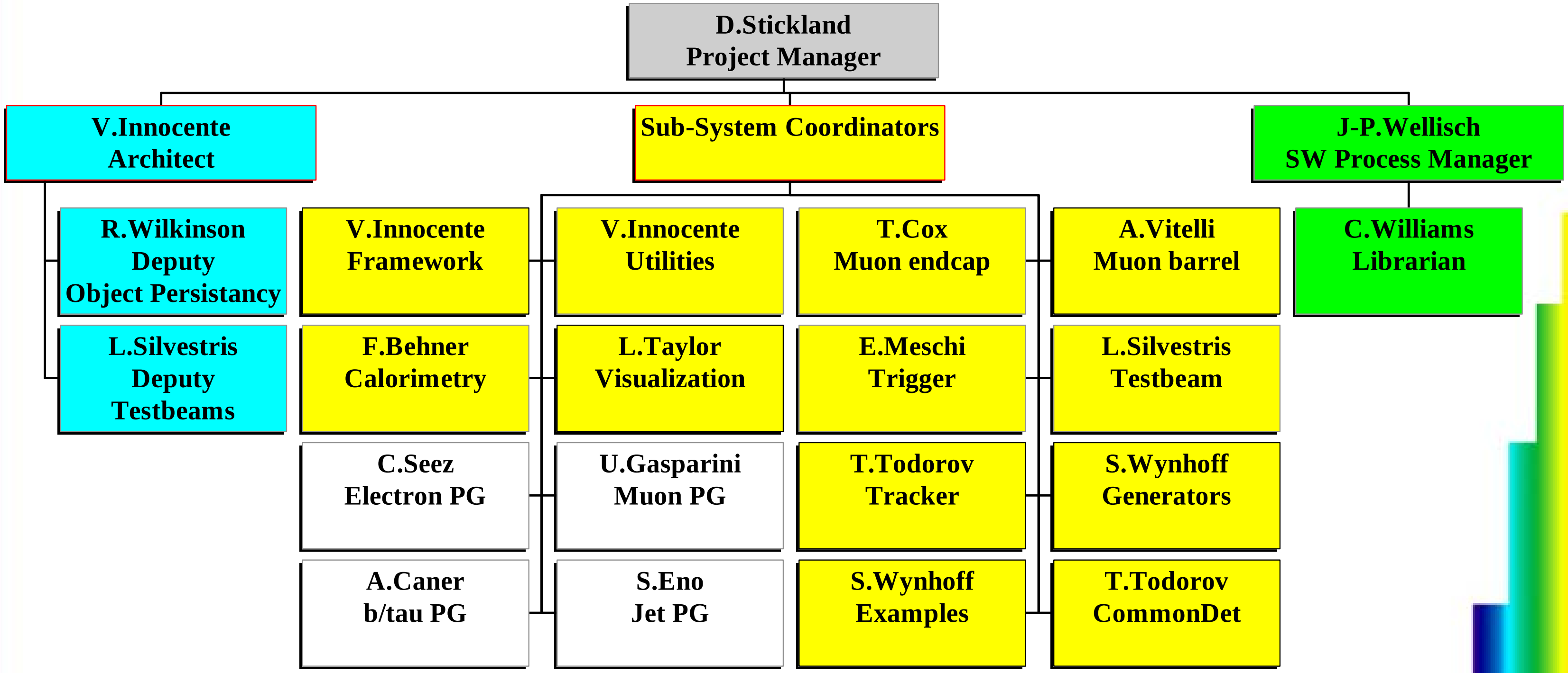
Size of CERN Software



Estimated Value of the main software packages
using the SlocCount tool (CoCoMo method)
see: http://www.dwheeler.com/sloccount/

	Lines of code	Person Years	Number Years	Number Developers	Total cost $ millions
Minuit	5913	1.29	0.59	2.19	0.174
Hbook	33415	7.96	1.18	6.76	1.075
Zebra	35058	8.38	1.21	6.97	1.135
Geant3	129727	33.09	2.02	16.34	4.471
PAW	284277	75.42	2.77	27.24	10.187
Geant4	339085	90.75	2.97	30.55	12.259
AliRoot	450782	122.38	3.33	36.77	16.531
ROOT	725969	201.83	4.02	50.15	27.265

CMS Offline Software



D.Stickland
Project Manager

V.Innocente
Architect

Sub-System Coordinators

J-P.Wellisch
SW Process Manager

R.Wilkinson
Deputy
Object Persistancy

L.Silvestris
Deputy
Testbeams

V.Innocente
Framework

F.Behner
Calorimetry

C.Seez
Electron PG

A.Caner
b/tau PG

V.Innocente
Utilities

L.Taylor
Visualization

U.Gasparini
Muon PG

S.Eno
Jet PG

T.Cox
Muon endcap

E.Meschi
Trigger

T.Todorov
Tracker

S.Wynhoff
Examples

A.Vitelli
Muon barrel

L.Silvestris
Testbeam

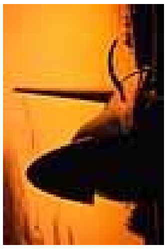
S.Wynhoff
Generators

T.Todorov
CommonDet

C.Williams
Librarian

Software Cop-Outs

- That's a feature, not a bug.
- If there are no questions, everyone must be happy.
- If there are no bug reports then noone is using it.
- We've lost the source code.
- We're too busy to document that.
- It must be a hardware problem.
- That could never fail -- don't bother testing for it.
- It's fixed, but is waiting for the next release cycle.



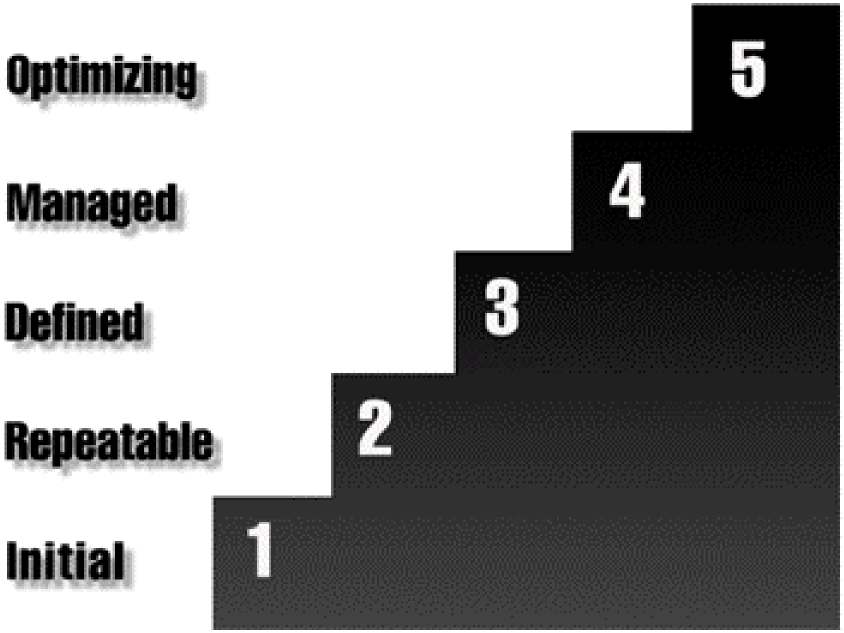
BRITISH AIRWAYS

The Software Process

- “The software process is the set of tools, methods and practices that are used to produce a software product.”

Watts S. Humphrey, Managing the Software Process

Capability Maturity Model



Software Development

The Mythical Man-Month

Frederic Brooks

- The mortal struggle of great beasts in the tar pits...

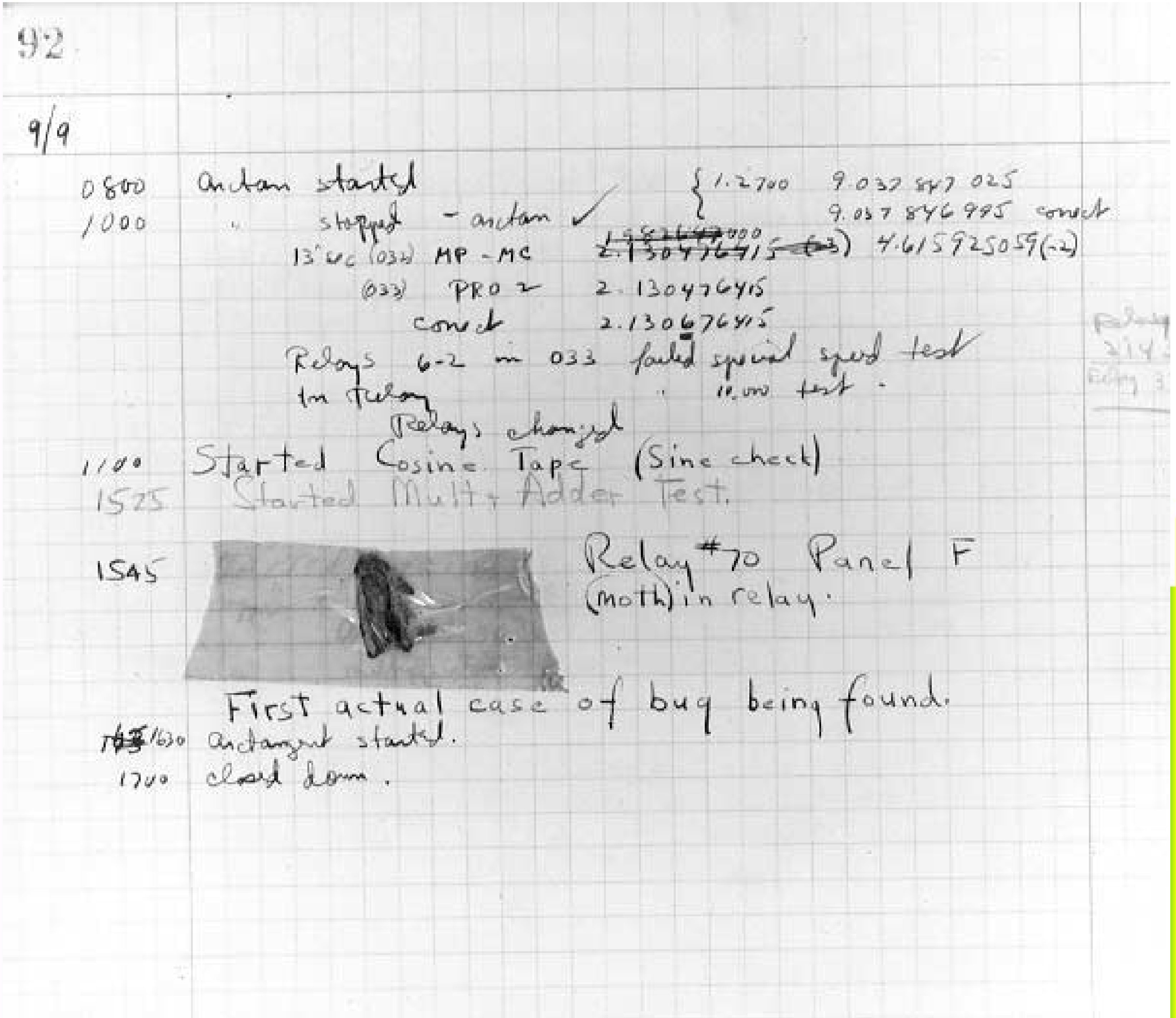
Software Scheduling

- 1/3 planning
- 1/6 coding
- 1/4 component testing
- 1/4 full system testing

From “The Mythical Man Month”

➢ The real cost is in maintenance & support!

The World's First Programmer



92

9/9

0800 arctan started {1.2700 9.037 847 025

1000 " stopped - arctan ✓ 9.037 846 995 correct

13" sec (032) MP - MC ~~1.982147000~~ ~~2.130476415 (-3)~~ 4.615925059(-2)

(033) PRO 2 2.130476415

correct 2.130676415

Relays 6-2 in 033 failed special speed test

in Relay " 11,000 test.

Relays changed

1100 Started Cosine Tape (Sine check)

1525 Started Mult + Adder Test.

1545 Relay #70 Panel F

(moth) in relay.

First actual case of bug being found.

~~1545~~ 1630 arctangent started.

1700 closed down.

Relay

214

Relay 3

Example I

The CERN Program Library:

CERNLIB

CERNLIB

- Arguably CERN’s most famous “product” prior to the Web
  - And it included **CERN** in the name…
- Written over nearly 40 years by at least as many authors
  - Try calculating the cost! **€100M** or more!
- Mainly Fortran, but some assembler, Pascal, C, …
- Used by virtually all HEP experiments world-wide, including those at the LHC!
- **No defined software process**
  - But steps in that direction…

CERNLIB: What is it?

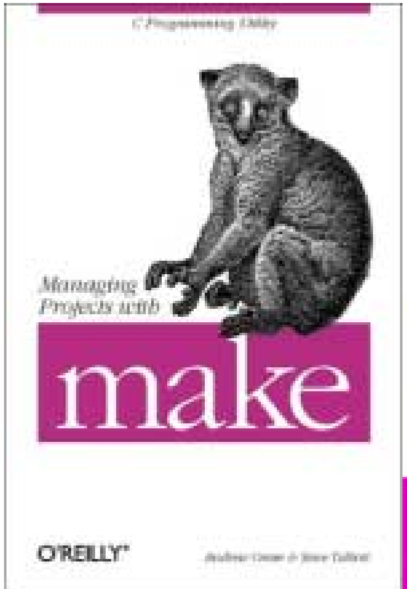
- Libraries (initially) and packages aimed at scientific computing
  - Histogramming, fitting, mathematical routines, graphics, analysis, detector simulation, event generators …
- “Tool kit” for physics software applications

CERNLIB cont.

- CERN Program Librarian – many incarnations
- Source code management: now CVS + cpp; previously home-grown cpp-equivalent
- Code conventions: must compile
- Build procedures: moved to *make* in 1990s
- Release procedures: old, pro & new areas
  - User testing of new area for weeks prior to release

Hall of fame: Make

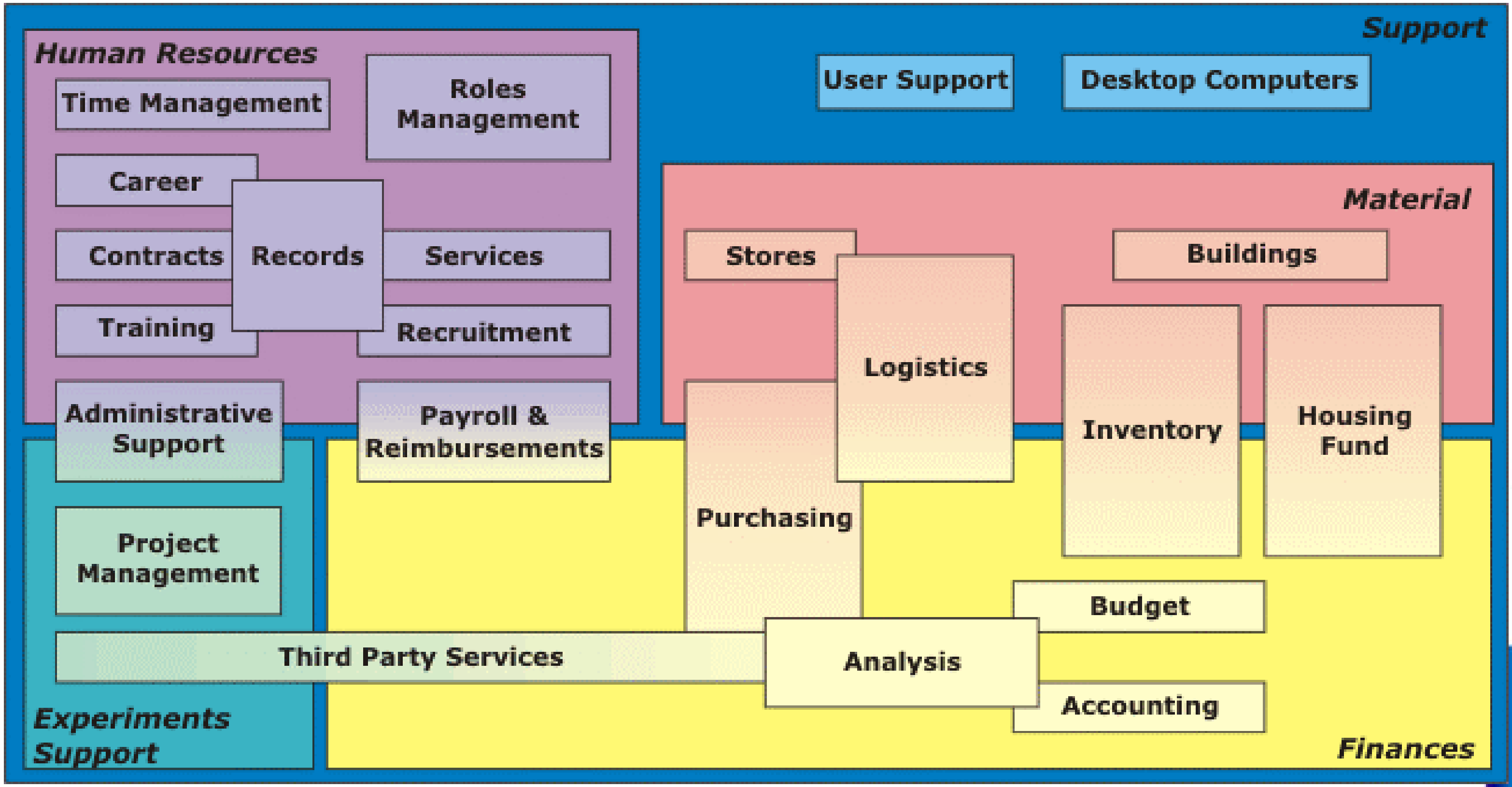
- Introduced to the world with Unix
  - Along with SCCS – “forerunner” of CVS
- Significant impact on software build process

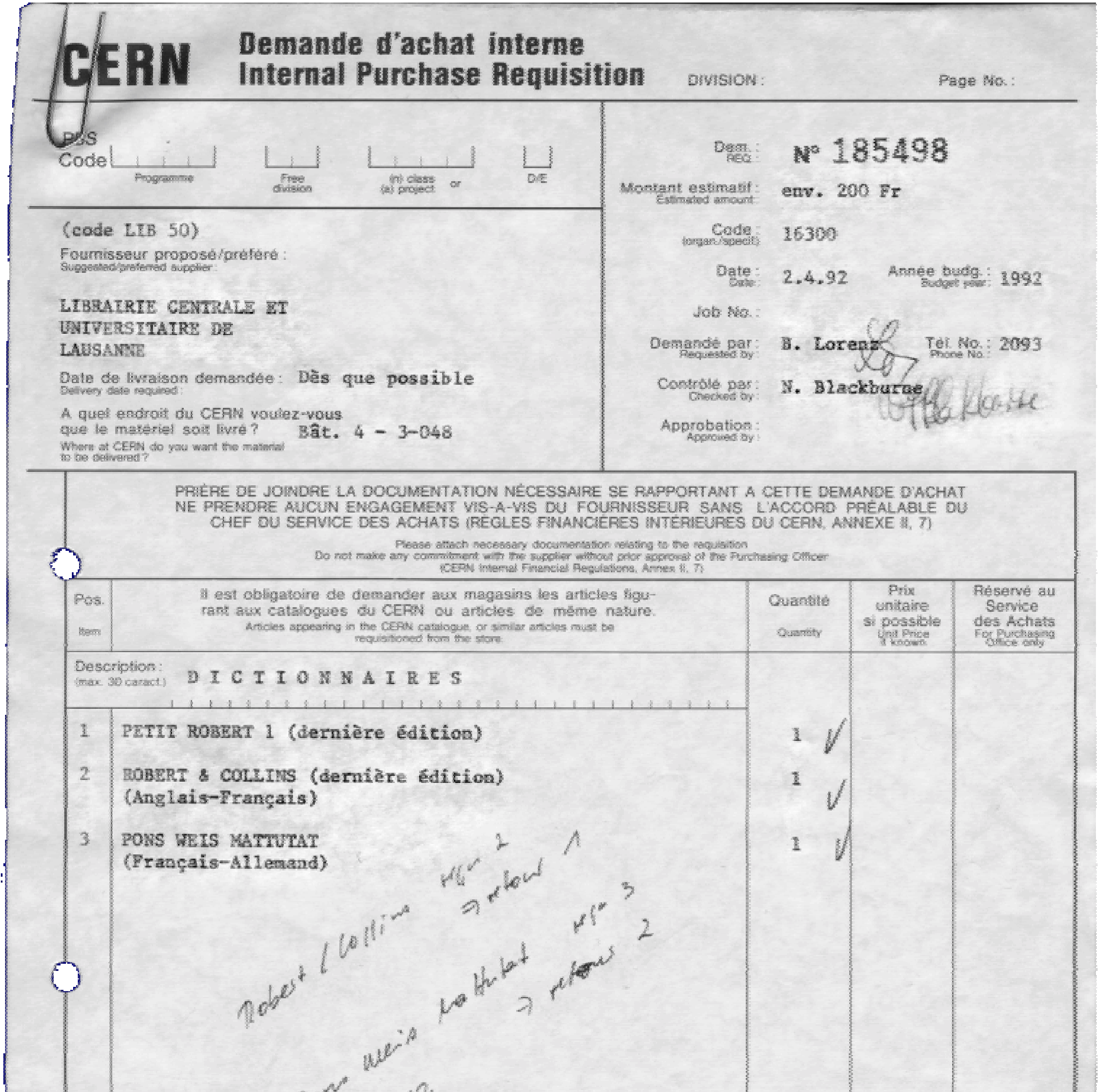


Example II

AIS Applications

AIS Applications





CERN **Demande d'achat interne**
Internal Purchase Requisition DIVISION : Page No. :

PBS Code

Programme | Free division | (n) class (e) project or | D/E

Dem. : REQ. : Nº 185498

Montant estimatif : Estimated amount : env. 200 Fr

(code LIB 50)

Code : (organ./specif.) 16300

Fournisseur proposé/préféré : Suggested/preferred supplier :

Date : Date : 2.4.92 Année budg. : Budget year : 1992

LIBRAIRIE CENTRALE ET
UNIVERSITAIRE DE
LAUSANNE

Job No. :

Demandé par : Requested by : B. Lorenz Tél. No. : Phone No. : 2093

Date de livraison demandée : Delivery date required : Dès que possible

Contrôlé par : Checked by : N. Blackburne

A quel endroit du CERN voulez-vous que le matériel soit livré ? Where at CERN do you want the material to be delivered ? Bât. 4 - 3-048

Approbation : Approved by :

PRIÈRE DE JOINDRE LA DOCUMENTATION NÉCESSAIRE SE RAPPORTANT A CETTE DEMANDE D'ACHAT
NE PRENDRE AUCUN ENGAGEMENT VIS-A-VIS DU FOURNISSEUR SANS L'ACCORD PRÉALABLE DU
CHEF DU SERVICE DES ACHATS (REGLES FINANCIÈRES INTÉRIEURES DU CERN, ANNEXE II, 7)

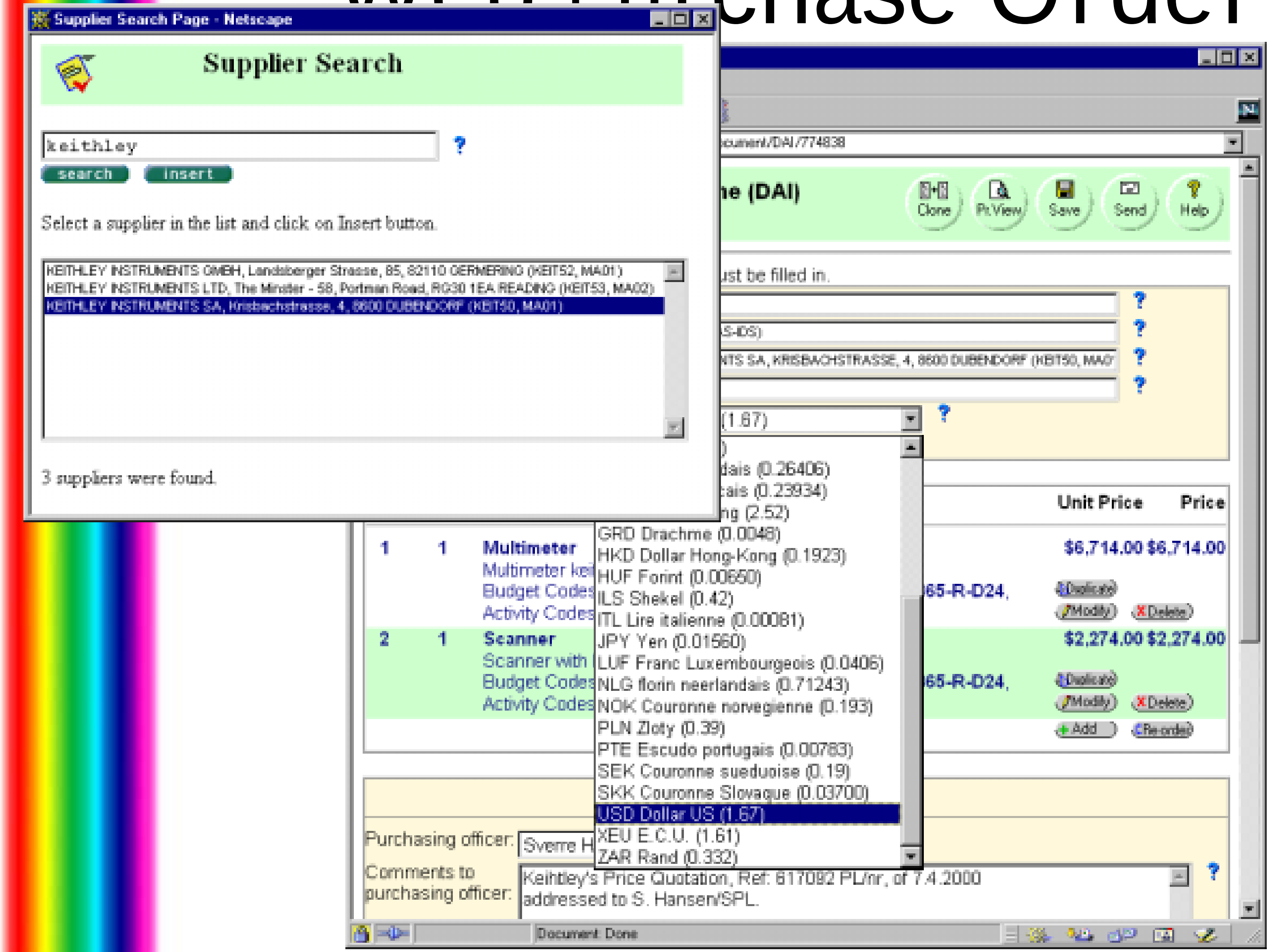
Please attach necessary documentation relating to the requisition
Do not make any commitment with the supplier without prior approval of the Purchasing Officer
(CERN Internal Financial Regulations, Annex II, 7)

Pos. Item	Il est obligatoire de demander aux magasins les articles figurant aux catalogues du CERN ou articles de même nature. Articles appearing in the CERN catalogue, or similar articles must be requisitioned from the store.	Quantité Quantity	Prix unitaire si possible Unit Price if known	Réservé au Service des Achats For Purchasing Office only
	Description : (max. 30 caract.) DICTIONNAIRES			
1	PETIT ROBERT 1 (dernière édition)	1 ✓		
2	ROBERT & COLLINS (dernière édition) (Anglais-Français)	1 ✓		
3	PONS WEIS MATTUTAT (Français-Allemand)	1 ✓		

Robert & Collins réf 2
→ retour 1

Pons Weis Mattutat réf 3
→ retour 2

Web Purchase Order



keithley

Select a supplier in the list and click on Insert button.

KEITHLEY INSTRUMENTS GMBH, Landsberger Strasse, 85, 82110 GERMERING (KEIT52, MA01)
KEITHLEY INSTRUMENTS LTD, The Minster - 58, Portman Road, RG30 1EA READING (KEIT53, MA02)
KEITHLEY INSTRUMENTS SA, Krisbachstrasse, 4, 8600 DUBENDORF (KEIT50, MA01)

3 suppliers were found.

ocument/DAI/774838

he (DAI)

ust be filled in.

S-IDS)

NTS SA, KRISBACHSTRASSE, 4, 8600 DUBENDORF (KEIT50, MA0

(1.67)

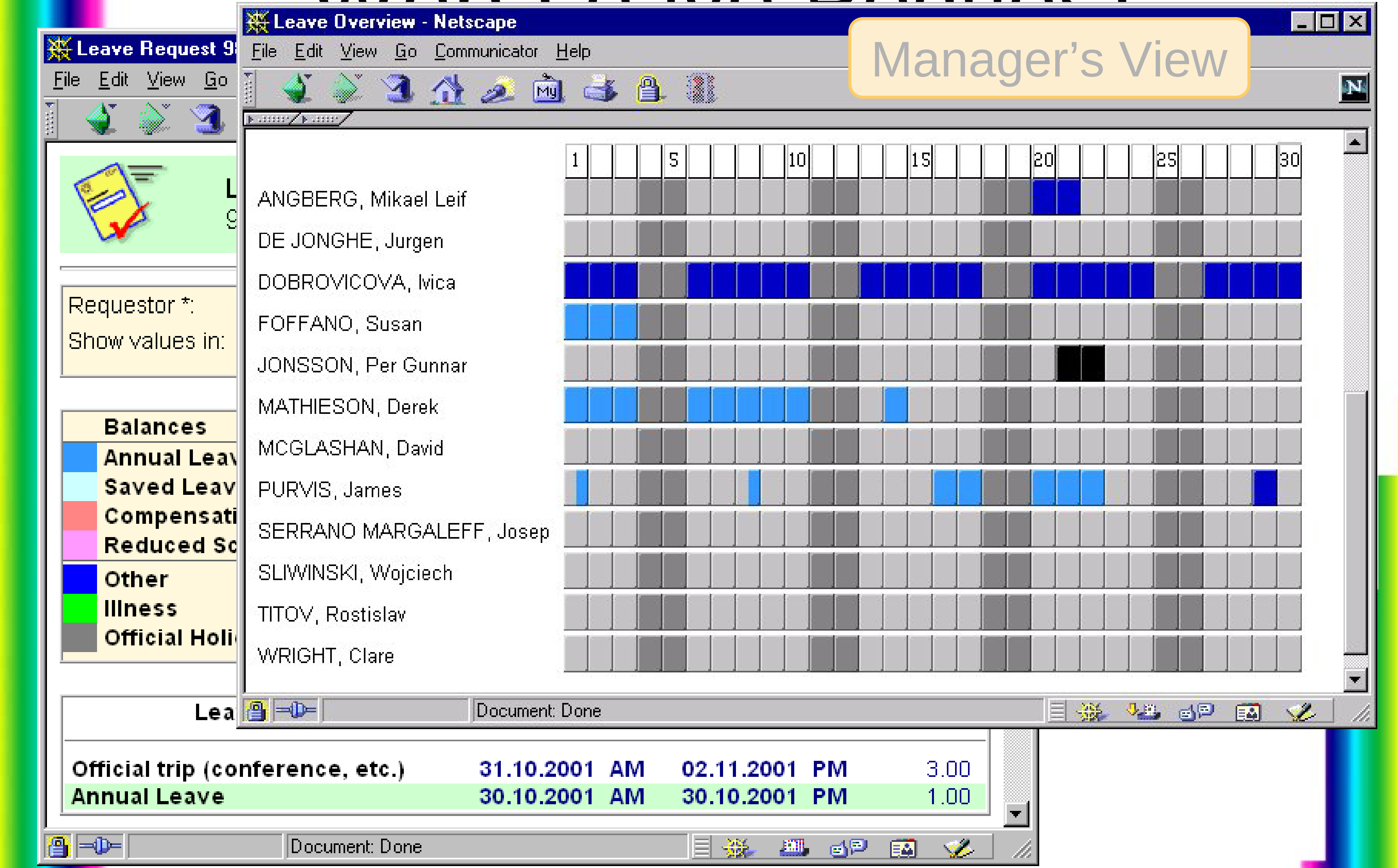
)
dais (0.26406)
cais (0.23934)
ng (2.52)
GRD Drachme (0.0048)
HKD Dollar Hong-Kong (0.1923)
HUF Forint (0.00650)
ILS Shekel (0.42)
ITL Lire italienne (0.00081)
JPY Yen (0.01560)
LUF Franc Luxembourgeois (0.0406)
NLG florin neerlandais (0.71243)
NOK Couronne norvegienne (0.193)
PLN Zloty (0.39)
PTE Escudo portugais (0.00783)
SEK Couronne sueduoise (0.19)
SKK Couronne Slovaque (0.03700)
USD Dollar US (1.67)
XEU E.C.U. (1.61)
ZAR Rand (0.332)

				Unit Price	Price
1	1	**Multimeter** Multimeter kei Budget Codes Activity Codes	65-R-D24,	$6,714.00	$6,714.00
2	1	**Scanner** Scanner with Budget Codes Activity Codes	65-R-D24,	$2,274.00	$2,274.00

Purchasing officer: Sverre H

Comments to purchasing officer: Keihtley's Price Quotation, Ref: 617092 PL/nr, of 7.4.2000 addressed to S. Hansen/SPL.

Web Leave Request



Manager's View

Leave Overview - Netscape
File Edit View Go Communicator Help
Leave Request 9
File Edit View Go

	1				5					10					15					20					25					30
ANGBERG, Mikael Leif				▒	▒						▒	▒						▒	▒	■	■				▒	▒				
DE JONGHE, Jurgen				▒	▒						▒	▒						▒	▒						▒	▒				
DOBROVICOVA, Ivica	■	■	■	▒	▒	■	■	■	■	■	▒	▒	■	■	■	■	■	▒	▒	■	■	■	■	■	▒	▒	■	■	■	■
FOFFANO, Susan	🟦	🟦	🟦	▒	▒						▒	▒						▒	▒						▒	▒				
JONSSON, Per Gunnar				▒	▒						▒	▒						▒	▒		⬛	⬛			▒	▒				
MATHIESON, Derek	🟦	🟦	🟦	▒	▒	🟦	🟦	🟦	🟦	🟦	▒	▒		🟦				▒	▒						▒	▒				
MCGLASHAN, David				▒	▒						▒	▒						▒	▒						▒	▒				
PURVIS, James	◨			▒	▒			◨			▒	▒				🟦	🟦	▒	▒	🟦	🟦	🟦			▒	▒			■	
SERRANO MARGALEFF, Josep				▒	▒						▒	▒						▒	▒						▒	▒				
SLIWINSKI, Wojciech				▒	▒						▒	▒						▒	▒						▒	▒				
TITOV, Rostislav				▒	▒						▒	▒						▒	▒						▒	▒				
WRIGHT, Clare				▒	▒						▒	▒						▒	▒						▒	▒				

Document: Done

L
9

Requestor *:
Show values in:

	Balances
🟦	Annual Leav
⬜	Saved Leav
🟥	Compensati
🟪	Reduced Sc
■	Other
🟩	Illness
▒	Official Holi

Lea

Official trip (conference, etc.)	31.10.2001	AM	02.11.2001	PM	3.00
Annual Leave	30.10.2001	AM	30.10.2001	PM	1.00

Document: Done

Standards and Inspections

- All Code must conform to coding standards
  - Informal Code Inspections
    - With follow-up

For more information see:

http://edh.cern.ch/CodingStandards

Example III

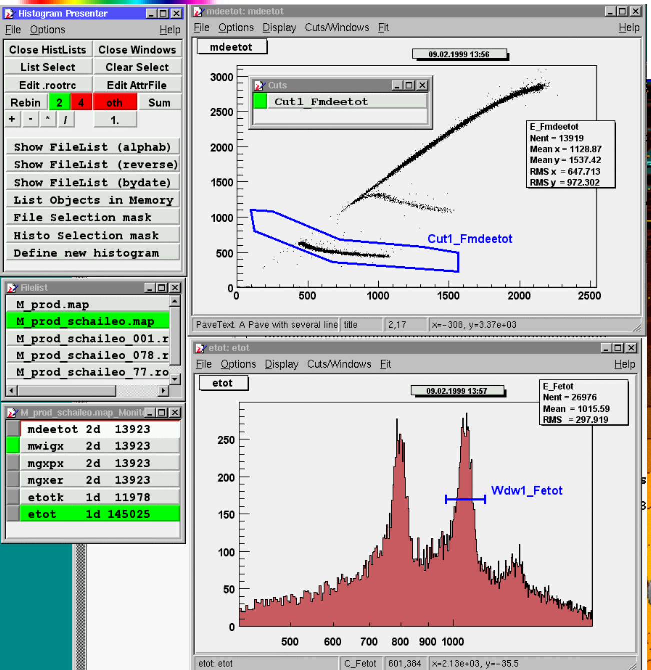
LCG Applications

Anaphe; Geant-4; ROOT; CMS, ...

http://wenaus.home.cern.ch/wenaus/peb-app/

LCG Applications

- Anaphe: a C++ replacement for CERNLIB
- GEANT4: a C++ detector simulation programme



M_prod.map
M_prod_schaileo.map
M_prod_schaileo_001.r
M_prod_schaileo_078.r
M_prod_schaileo_77.ro

M_prod_schaileo.map_Monit

mdeetot	2d	13923
mwigx	2d	13923
mgxpx	2d	13923
mgxer	2d	13923
etotk	1d	11978
etot	1d	145025

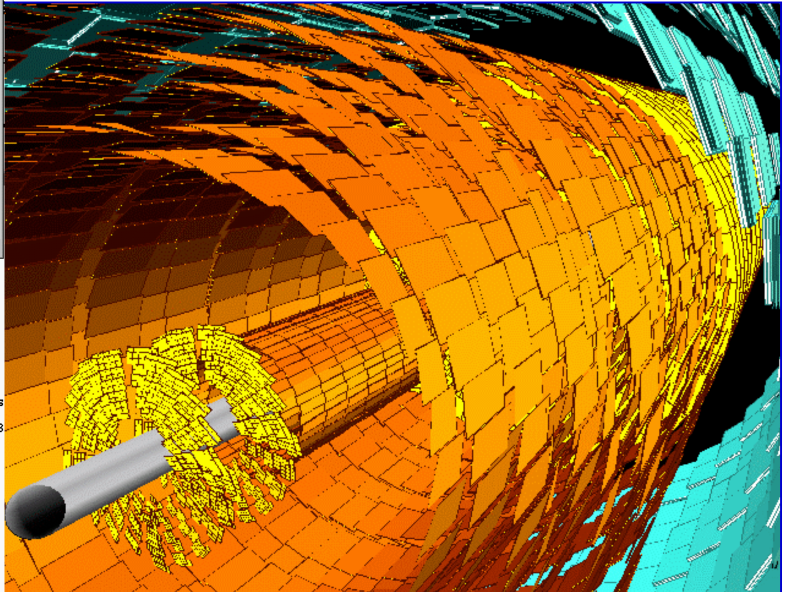
```
E_Fmdeetot
Nent = 13919
Mean x = 1128.87
Mean y = 1537.42
RMS x  = 647.713
RMS y  = 972.302
```

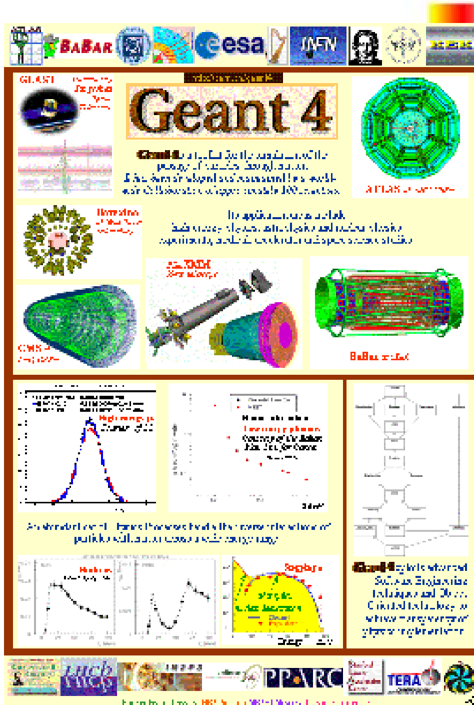
PaveText A Pave with several line | title | 2,17 | x=-308, y=3.37e+03

```
E_Fetot
Nent = 26976
Mean = 1015.59
RMS  = 297.919
```

etot: etot | C_Fetot | 601,384 | x=2.13e+03, y=-35.5

IGUANA





Anaphe releases

- Do not use CERNLIB-style old / pro / new
  - Limited flexibility (esp. with shared libs)
  - Bad "recognisability" : "*pro*" in outside institutes may differ from "*pro*" at CERN (and even within institutes)
  - ➔ use version *numbers*
- Version numbers for each package
  - Component based architecture allows for (semi-) independent development
  - Version numbers in library names
    - ***lib<pkg>.<vers>.so*** (link: ***lib<pkg>.so -> lib<pkg>.<vers>.so***)
- Coherent set of versioned packages as "release"

Geant4 releases

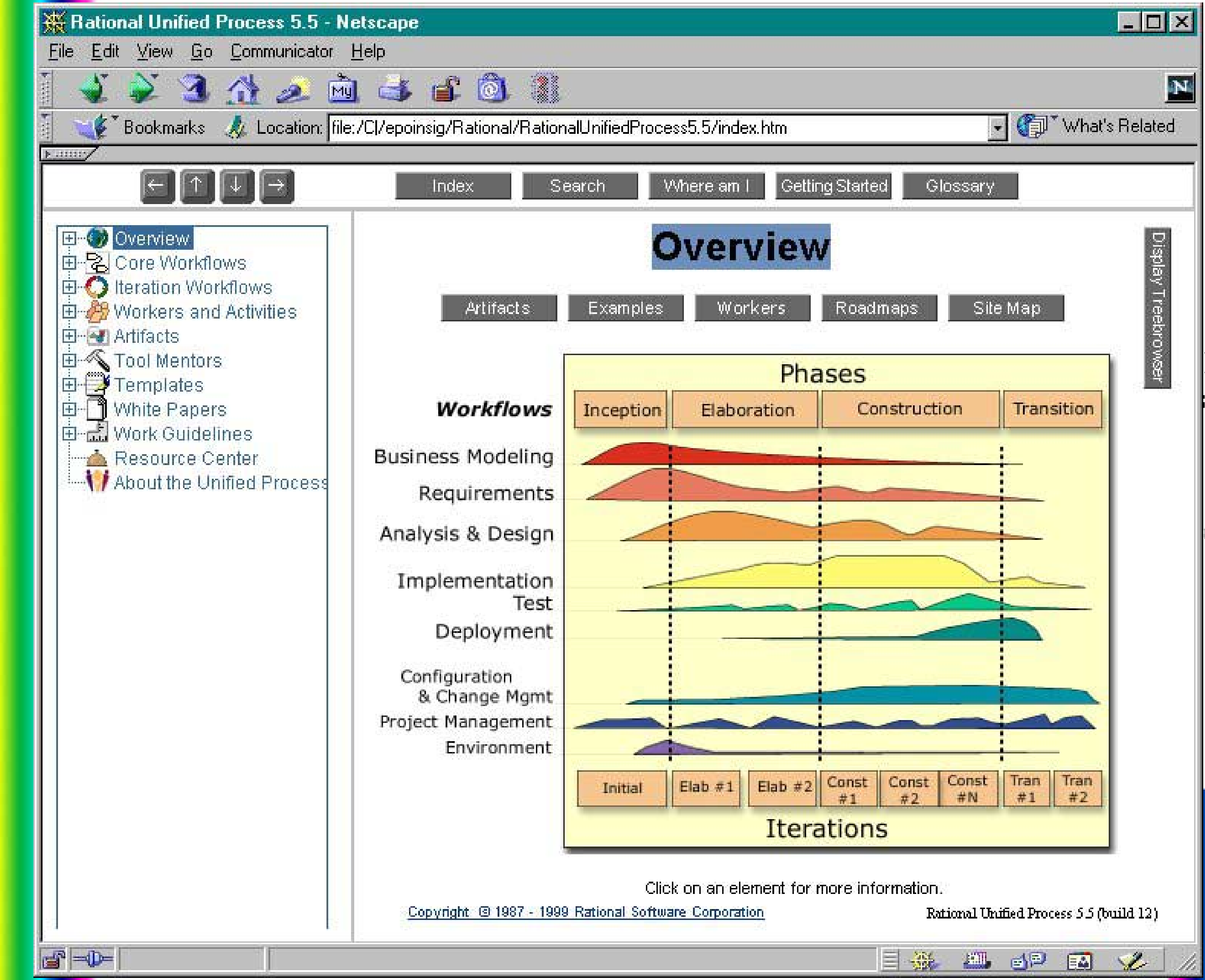
- Major releases
  - include major changes and updates, including public interface changes. May require porting of users' code
  - represented by major revision number *XX* in *XX.YY*
- Minor releases
  - include updates, bug-fixes and new features NOT affecting public interfaces in the code
  - represented by minor revision number *YY* in *XX.YY*
- Public patches
  - include exclusively bug-fixes to a public release
- Development releases
  - include "state-of-art" development and fixes not yet submitted to acceptance as public supported release

Software development: the traditional approach

- Correcting software errors very expensive: errors should not be in the code in the first place
  - ***get it right the first time***
- ***Sound traditional engineering techniques*** must be applied
- This leads to **Big-Design-Up-Front**
  - Waterfall, SEI CMM, and other techniques were developed for this purpose
- They are *High ceremony processes*

An analogy: building a skyscraper

- Detailed architectural and structural designs are needed
- Specialized architects and engineers create the design
- The building is made by technicians and workers, following the design
- The skyscraper is ***made right the first time!***



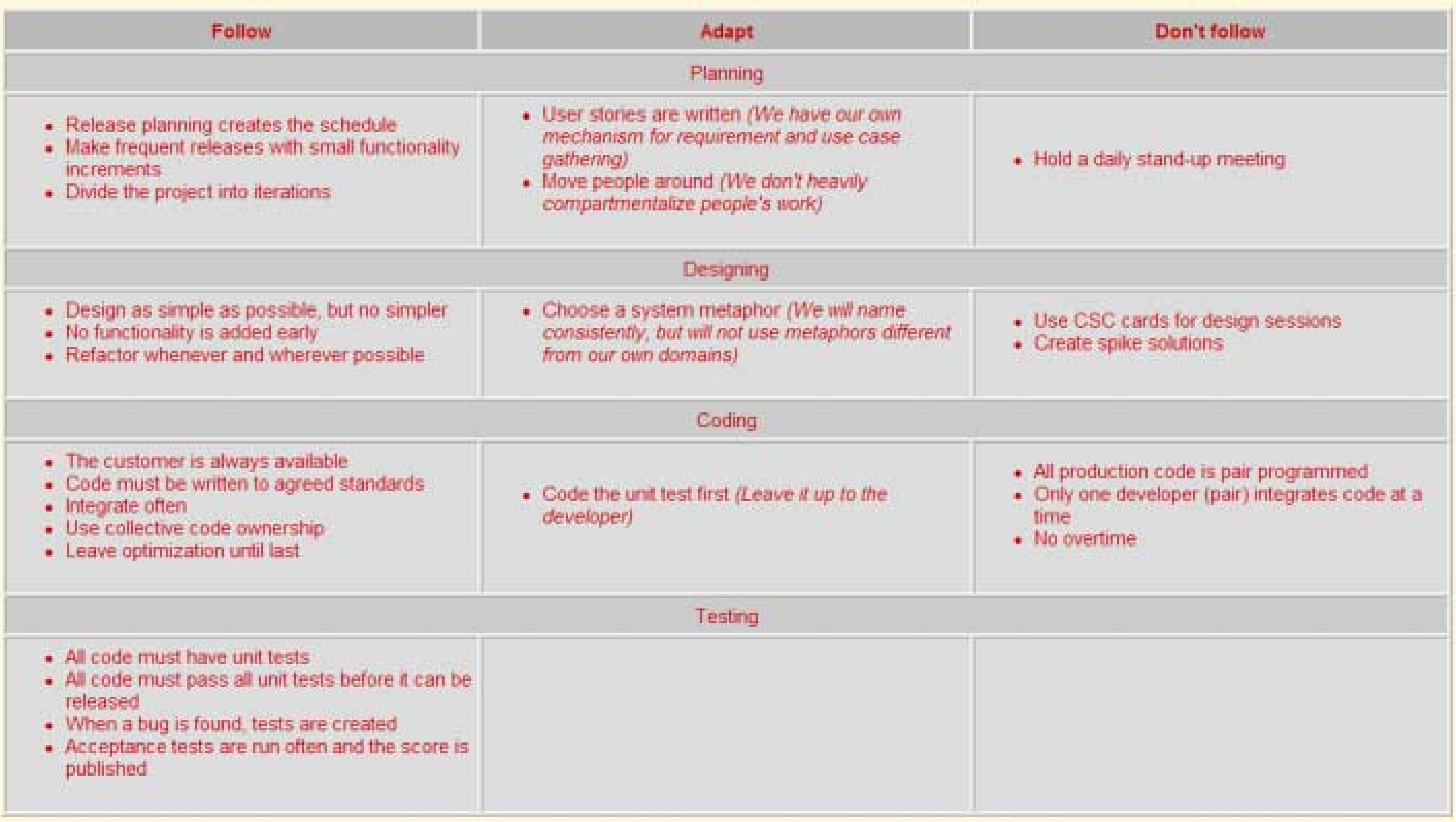
Overview

Click on an element for more information.

Copyright © 1987 - 1999 Rational Software Corporation

Rational Unified Process 5.5 (build 12)

What should we take from XP?



Follow	Adapt	Don't follow
Planning		
Release planning creates the schedule Make frequent releases with small functionality increments Divide the project into iterations	User stories are written *(We have our own mechanism for requirement and use case gathering)* Move people around *(We don't heavily compartmentalize people's work)*	Hold a daily stand-up meeting
Designing		
Design as simple as possible, but no simpler No functionality is added early Refactor whenever and wherever possible	Choose a system metaphor *(We will name consistently, but will not use metaphors different from our own domains)*	Use CSC cards for design sessions Create spike solutions
Coding		
The customer is always available Code must be written to agreed standards Integrate often Use collective code ownership Leave optimization until last	Code the unit test first *(Leave it up to the developer)*	All production code is pair programmed Only one developer (pair) integrates code at a time No overtime
Testing		
All code must have unit tests All code must pass all unit tests before it can be released When a bug is found, tests are created Acceptance tests are run often and the score is published		

Pair Programming

What should we take from RUP?

- We should follow all the suggested "best practices"
  - Develop iteratively.
  - Manage requirements.
  - Use component architectures.
  - Model visually.
  - Verify quality.
  - Control changes.

Summary

Summary

Producing high-quality software is:

- Far from easy
- Far from cheap
- Still not a solved problem

Lecture III

- Computing at CERN Today
- Software at CERN Today

➢ **The future & LHC Computing**

Homework

Exercise II

- What will the CERN Computing environment look like in 10 years?
- Hint: some of the key elements exist today, albeit possibly in a different flavour.

End Lecture I I