

# Developing successful common software projects some lessons

René Brun /CERN  
HSF workshop, SLAC  
January 2015

# Success in developing software

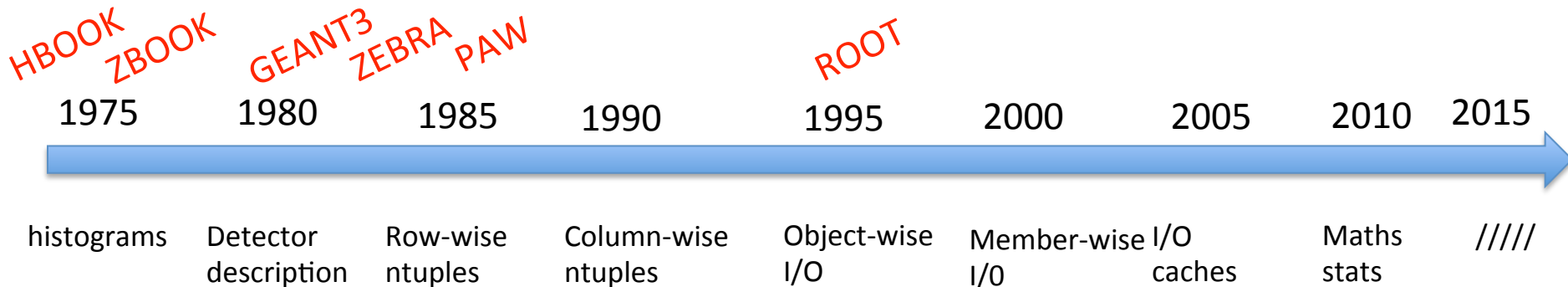
- Development of tools and libs
  - Attracting a substantial fraction of HEP users and other fields too.
  - With a long life time and gateways for evolution.
  - Influencing the way experiment software is built.
- Enforcing some basic principles
  - To motivate developers
  - To attract new contributors
- Accepting competition

# Key elements

- Understand the requirements by being strongly involved in the field for which you propose a solution.
- Propose an architecture with a solid&light backbone.
- Develop appropriate interfaces by testing and prototyping.
- Rapid prototyping to demonstrate the main ideas and get support.
- Get early users with instantaneous support.
- Be trusted. It takes time.
- Understand basics of Psychology and Sociology

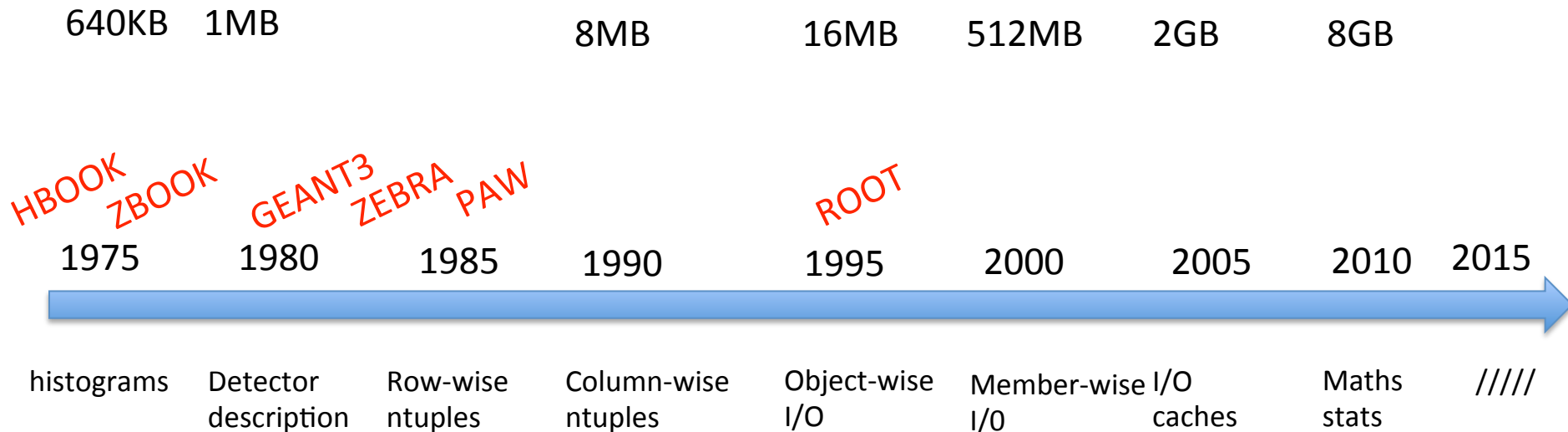
# Did you say “Backbone” ?

- Like human bodies, software bodies need a backbone and a theme supporting the most common data structures and operations.
  - I/O, file structure, self-describing objects
  - Common interfaces: browsers, queries, interpreters.



# Did you say “fat”?

- Size of systems
- Size of machines
- Size of experiments
- Plug-in and unplug-in managers



# Size of systems and interfaces

- Fat classes vs too many classes
- Being lost in the hyper space



# Understanding user support

- Many types of users: framework to end users
- Prompt answers to any question is a must.
- Understanding messages from users, when the same problem(s) reappear.
- From mailing lists to open forum.
- ROOT forum today:
  - 8252 members for 25000 estimated users
  - 80675 posts since Oct2003 (16 per day)

# Packaging/Distribution/Validation

- Develop portable software
- Build system must be simple to develop and maintain.
- Testing must be systematic, with dashboards
- Tests must correspond to realistic use cases.
- Coding conventions checkers and code quality checkers must be run at least daily.
- Towards “Aps” style installation/update



# Understanding failures

- Many projects appear and disappear.
- When a project disappears, there is in general no post-mortem analysis.
- One may learn more from a failure than a success!

# Remember

- Google, Facebook or CandyCrush were not created by a committee !
- Creativity is often in contradiction with management or “bureaucratical” behaviors..
- Successful projects are implemented by motivated and enthusiastic members.
- Enthusiasm and motivation are implicit if members see a medium and long term perspective.