

# Développement collaboratif libre d'applications métier

## Étude de cas: Sage pour les Mathématiques

Nicolas M. Thiéry

Institut Universitaire de Technologie d'Orsay  
Laboratoire de Mathématiques d'Orsay  
Université Paris-Sud 11



Linux Party, 5 février 2011  
École Mohammadia d'Ingénieurs  
Rabat, Maroc

# Le projet \*-Combinat : tout a commencé là



# Le projet \*-Combinat : 1

Nicolas

20k

Le projet \*-Combinat :  $1+1 =$

Nicolas

20k

Florent

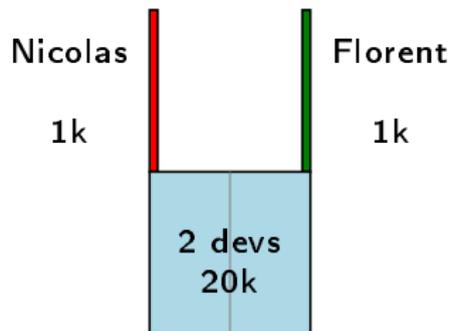
20k

Le projet \*-Combinat :  $1+1 =$

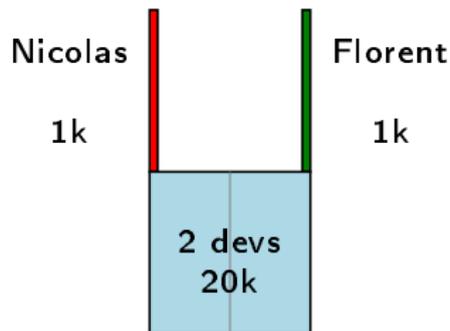
Nicolas  
20k

Florent  
20k

Le projet \*-Combinat :  $1+1 = 1.1$

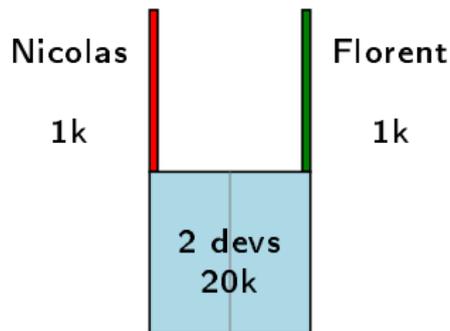


# Le projet \*-Combinat : $1+1 = 1.1$



- 95% des besoins de développements sont génériques

## Le projet \*-Combinat : $1+1 = 1.1$



- 95% des besoins de développements sont génériques
- Opportunité de partage et mutualisation

# Le projet \*-Combinat : [combinat.sagemath.org](http://combinat.sagemath.org)

## Mission

*«Améliorer MuPAD/Sage comme boîte à outils extensible pour l'exploration informatique en combinatoire algébrique, en fédérant et mutualisant les efforts de développements des chercheurs»*

# Le projet \*-Combinat : [combinat.sagemath.org](http://combinat.sagemath.org)

## Mission

*«Améliorer MuPAD/Sage comme boîte à outils extensible pour l'exploration informatique en combinatoire algébrique, en fédérant et mutualisant les efforts de développements des chercheurs»*

## Stratégie

- Licence libre pour partager avec le plus grand nombre  
*En restant pragmatique dans les collaborations*
- Développement décentralisé et international  
*Garantie d'indépendance vis-à-vis des tutelles*
- Développé par des chercheurs pour des chercheurs  
*Avec un usage plus large en vue*
- Coeur du développement par des permanents  
*Les doctorants se concentrent sur leurs propres besoins*
- Chaque ligne de code justifiée par un projet de recherche  
*Avec une vision à long terme* (développement agile)
- Inspiration des informaticiens :  
*Concepts et méthodologies de programmation*  
*Outils de développement coopératif*

## Le projet \*-Combinat : 10 ans après

### En quelques chiffres

- MuPAD-Combinat : 115k lignes de MuPAD, 15k lignes de C++, 32k lignes de tests, 600 pages de doc
- Sage-Combinat : 300 tickets / 100k lignes dans Sage
- Financements : ANR, PEPS, NSF, Google Summer of Code

## Le projet \*-Combinat : 10 ans après

### En quelques chiffres

- MuPAD-Combinat : 115k lignes de MuPAD, 15k lignes de C++, 32k lignes de tests, 600 pages de doc
- Sage-Combinat : 300 tickets / 100k lignes dans Sage
- Financements : ANR, PEPS, NSF, Google Summer of Code
- *70+ articles de recherche*

# Le projet \*-Combinat : 10 ans après

## En quelques chiffres

- MuPAD-Combinat : 115k lignes de MuPAD, 15k lignes de C++, 32k lignes de tests, 600 pages de doc
- Sage-Combinat : 300 tickets / 100k lignes dans Sage
- Financements : ANR, PEPS, NSF, Google Summer of Code
- *70+ articles de recherche*

## Une communauté :

Nicolas Borie, Daniel Bump, Jason Bandlow, Adrien Boussicault, Frédéric Chapoton, Vincent Delecroix, Paul-Olivier Dehaye, Tom Denton, François Descouens, Dan Drake, Teresa Gomez Diaz, Valentin Feray, Mike Hansen, Ralf Hemmecke, Florent Hivert, Brant Jones, Sébastien Labbé, Yann Laigle-Chapuy, Éric Laugerotte, Patrick Lemeur, Andrew Mathas, Xavier Molinero, Thierry Monteil, Olivier Mallet, Gregg Musiker, Jean-Christophe Novelli, Janvier Nzeutchap, Steven Pon, Viviane Pons, Franco Saliola, Anne Schilling, Mark Shimozono, Christian Stump, Lenny Tevlin, Nicolas M. Thiéry, Justin Walker, Qiang Wang, Mike Zabrocki, ...

# Le projet \*-Combinat : 10 ans après

## En quelques chiffres

- MuPAD-Combinat : 115k lignes de MuPAD, 15k lignes de C++, 32k lignes de tests, 600 pages de doc
- Sage-Combinat : 300 tickets / 100k lignes dans Sage
- Financements : ANR, PEPS, NSF, Google Summer of Code
- *70+ articles de recherche*

## Une communauté :

Nicolas Borie, Daniel Bump, Jason Bandlow, Adrien Boussicault, Frédéric Chapoton, Vincent Delecroix, Paul-Olivier Dehaye, Tom Denton, François Descouens, Dan Drake, Teresa Gomez Diaz, Valentin Feray, Mike Hansen, Ralf Hemmecke, Florent Hivert, Brant Jones, Sébastien Labbé, Yann Laigle-Chapuy, Éric Laugerotte, Patrick Lemeur, Andrew Mathas, Xavier Molinero, Thierry Monteil, Olivier Mallet, Gregg Musiker, Jean-Christophe Novelli, Janvier Nzeutchap, Steven Pon, Viviane Pons, Franco Saliola, Anne Schilling, Mark Shimozono, Christian Stump, Lenny Tevlin, Nicolas M. Thiéry, Justin Walker, Qiang Wang, Mike Zabrocki, ...

*Et vous ?*

# Le projet Sage

## Mission

« *Créer une alternative libre et viable à  
Maple<sup>TM</sup>, Mathematica<sup>TM</sup>, Magma<sup>TM</sup> et Matlab<sup>TM</sup> ...*

# Le projet Sage

## Mission

*« Créer une alternative libre et viable à  
Maple<sup>TM</sup>, Mathematica<sup>TM</sup>, Magma<sup>TM</sup> et Matlab<sup>TM</sup> ...  
et une communauté accueillante d'utilisateurs et de développeurs »*

# Le projet Sage

## Mission

« *Créer une alternative libre et viable à  
Maple<sup>TM</sup>, Mathematica<sup>TM</sup>, Magma<sup>TM</sup> et Matlab<sup>TM</sup> ...  
et une communauté accueillante d'utilisateurs et de développeurs* »

## Stratégie de développement

- Entièrement libre (GPL)

# Le projet Sage

## Mission

« *Créer une alternative libre et viable à  
Maple<sup>TM</sup>, Mathematica<sup>TM</sup>, Magma<sup>TM</sup> et Matlab<sup>TM</sup> ...  
et une communauté accueillante d'utilisateurs et de développeurs* »

## Stratégie de développement

- Entièrement libre (GPL)
- Développé par et pour une communauté d'utilisateurs

# Le projet Sage

## Mission

*« Créer une alternative libre et viable à  
Maple<sup>TM</sup>, Mathematica<sup>TM</sup>, Magma<sup>TM</sup> et Matlab<sup>TM</sup> ...  
et une communauté accueillante d'utilisateurs et de développeurs »*

## Stratégie de développement

- Entièrement libre (GPL)
- Développé par et pour une communauté d'utilisateurs
- « Construire la voiture plutôt que réinventer la roue »  
Atlas, GAP, GMP, Linbox, Maxima, MPFR, PARI/GP,  
NetworkX, NTL, Numpy/Scipy, Singular, Symmetrica, ...

# Le projet Sage

## Mission

*« Créer une alternative libre et viable à  
Maple<sup>TM</sup>, Mathematica<sup>TM</sup>, Magma<sup>TM</sup> et Matlab<sup>TM</sup> ...  
et une communauté accueillante d'utilisateurs et de développeurs »*

## Stratégie de développement

- Entièrement libre (GPL)
- Développé par et pour une communauté d'utilisateurs
- « Construire la voiture plutôt que réinventer la roue »  
Atlas, GAP, GMP, Linbox, Maxima, MPFR, PARI/GP,  
NetworkX, NTL, Numpy/Scipy, Singular, Symmetrica, ...
- Basé sur un langage de programmation standard (Python)

# Le projet Sage

## Mission

*« Créer une alternative libre et viable à  
Maple<sup>TM</sup>, Mathematica<sup>TM</sup>, Magma<sup>TM</sup> et Matlab<sup>TM</sup> ...  
et une communauté accueillante d'utilisateurs et de développeurs »*

## Stratégie de développement

- Entièrement libre (GPL)
- Développé par et pour une communauté d'utilisateurs
- « Construire la voiture plutôt que réinventer la roue »  
Atlas, GAP, GMP, Linbox, Maxima, MPFR, PARI/GP,  
NetworkX, NTL, Numpy/Scipy, Singular, Symmetrica, ...
- Basé sur un langage de programmation standard (Python)
- Modèle de développement agile

# Le projet Sage

## Mission

*« Créer une alternative libre et viable à  
Maple<sup>TM</sup>, Mathematica<sup>TM</sup>, Magma<sup>TM</sup> et Matlab<sup>TM</sup> ...  
et une communauté accueillante d'utilisateurs et de développeurs »*

## Stratégie de développement

- Entièrement libre (GPL)
- Développé par et pour une communauté d'utilisateurs
- « Construire la voiture plutôt que réinventer la roue »  
Atlas, GAP, GMP, Linbox, Maxima, MPFR, PARI/GP,  
NetworkX, NTL, Numpy/Scipy, Singular, Symmetrica, ...
- Basé sur un langage de programmation standard (Python)
- Modèle de développement agile
- Prosélytisme actif (et critique !)

## Mini historique de Sage

- *1999-2005* : William Stein écrit plus de 25,000 lignes de code Magma pour sa recherche. Il réalise que c'est un mauvais investissement à long terme car il ne peut ni voir ni changer l'intérieur de Magma.
- *Fev. 2005* : Sage 0.1, une bibliothèque Python liant ensemble PARI, Maxima, Python, Singular, GAP.

## Mini historique de Sage

- *1999-2005* : William Stein écrit plus de 25,000 lignes de code Magma pour sa recherche. Il réalise que c'est un mauvais investissement à long terme car il ne peut ni voir ni changer l'intérieur de Magma.
- *Fev. 2005* : Sage 0.1, une bibliothèque Python liant ensemble PARI, Maxima, Python, Singular, GAP.
- *Fev. 2006* : Sage 1.0  
Sage Days 1, San Diego, 10 participants ?

## Mini historique de Sage

- *1999-2005* : William Stein écrit plus de 25,000 lignes de code Magma pour sa recherche. Il réalise que c'est un mauvais investissement à long terme car il ne peut ni voir ni changer l'intérieur de Magma.
- *Fev. 2005* : Sage 0.1, une bibliothèque Python liant ensemble PARI, Maxima, Python, Singular, GAP.
- *Fev. 2006* : Sage 1.0  
Sage Days 1, San Diego, 10 participants ?
- *Fev. 2010* : Sage 4.4.2  
Sage Days 20, Luminy (France), 120 participants

## Mini historique de Sage

- *1999-2005* : William Stein écrit plus de 25,000 lignes de code Magma pour sa recherche. Il réalise que c'est un mauvais investissement à long terme car il ne peut ni voir ni changer l'intérieur de Magma.
- *Fev. 2005* : Sage 0.1, une bibliothèque Python liant ensemble PARI, Maxima, Python, Singular, GAP.
- *Fev. 2006* : Sage 1.0  
Sage Days 1, San Diego, 10 participants ?
- *Fev. 2010* : Sage 4.4.2  
Sage Days 20, Luminy (France), 120 participants
- *Fev. 2011* : Sage 4.6.1
- 200 contributeurs dans le monde entier
- 10000 utilisateurs ?
- *Financements* (postdocs, ateliers, matériel) : NSF, ANR, CNRS, Universités et instituts, Google, Microsoft Research, ...

## Quelques livres libres

- *Calcul Mathématique avec Sage*

Alexandre Casamayou, Guillaume Connan

Thierry Dumont, Laurent Fousse

François Maltey, Matthias Meulien

Marc Mezzarobba, Clément Pernet

Nicolas M. Thiéry, Paul Zimmermann

<http://sagebook.gforge.inria.fr/>

July 2010 : 1.0 en ligne    August 2011 : imprimé

Successesseur de : *Calcul formel, mode d'emploi*

Dumas, Gomez, Salvy, Zimmermann

- *The Sage tutorial*

David Joyner, William Stein et al.

- *A First Course in Linear Algebra*

Robert Beezer

## Trois clefs de la viabilité

Généricité



Évaluation

Dépôt

# Généricité

Un calcul typique en combinatoire algébrique nécessite :

- Un peu de combinatoire usuelle
- Un peu d'algèbre linéaire
- Un peu de théorie des groupes
- Un peu de calcul formel
- Un peu de ...

# Généricité

Un calcul typique en combinatoire algébrique nécessite :

- Un peu de combinatoire usuelle
- Un peu d'algèbre linéaire
- Un peu de théorie des groupes
- Un peu de calcul formel
- Un peu de ...
- Et cette fine couche de ma propre poudre magique

## Généricité

Un calcul typique en combinatoire algébrique nécessite :

- Un peu de combinatoire usuelle
- Un peu d'algèbre linéaire
- Un peu de théorie des groupes
- Un peu de calcul formel
- Un peu de ...
- Et cette fine couche de ma propre poudre magique

*Un logiciel de combinatoire n'est jamais complet*

## Généricité

Un calcul typique en combinatoire algébrique nécessite :

- Un peu de combinatoire usuelle
- Un peu d'algèbre linéaire
- Un peu de théorie des groupes
- Un peu de calcul formel
- Un peu de ...
- Et cette fine couche de ma propre poudre magique

*Un logiciel de combinatoire n'est jamais complet*

*Je veux être un architecte, me concentrer sur ma poudre magique*

## Généricité

Un calcul typique en combinatoire algébrique nécessite :

- Un peu de combinatoire usuelle
- Un peu d'algèbre linéaire
- Un peu de théorie des groupes
- Un peu de calcul formel
- Un peu de ...
- Et cette fine couche de ma propre poudre magique

*Un logiciel de combinatoire n'est jamais complet*

*Je **doit** être un architecte, me concentrer sur ma poudre magique*

# Généricité

Un calcul typique en combinatoire algébrique nécessite :

- Un peu de combinatoire usuelle
- Un peu d'algèbre linéaire
- Un peu de théorie des groupes
- Un peu de calcul formel
- Un peu de ...
- Et cette fine couche de ma propre poudre magique

*Un logiciel de combinatoire n'est jamais complet*

*Je **doit** être un architecte, me concentrer sur ma poudre magique*

Mais est-ce possible ?

Moi

1

1k

## \*-Combinat

Moi

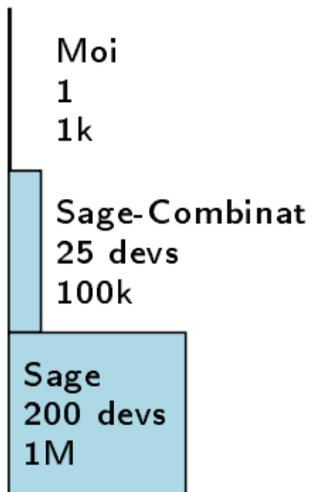
1

1k

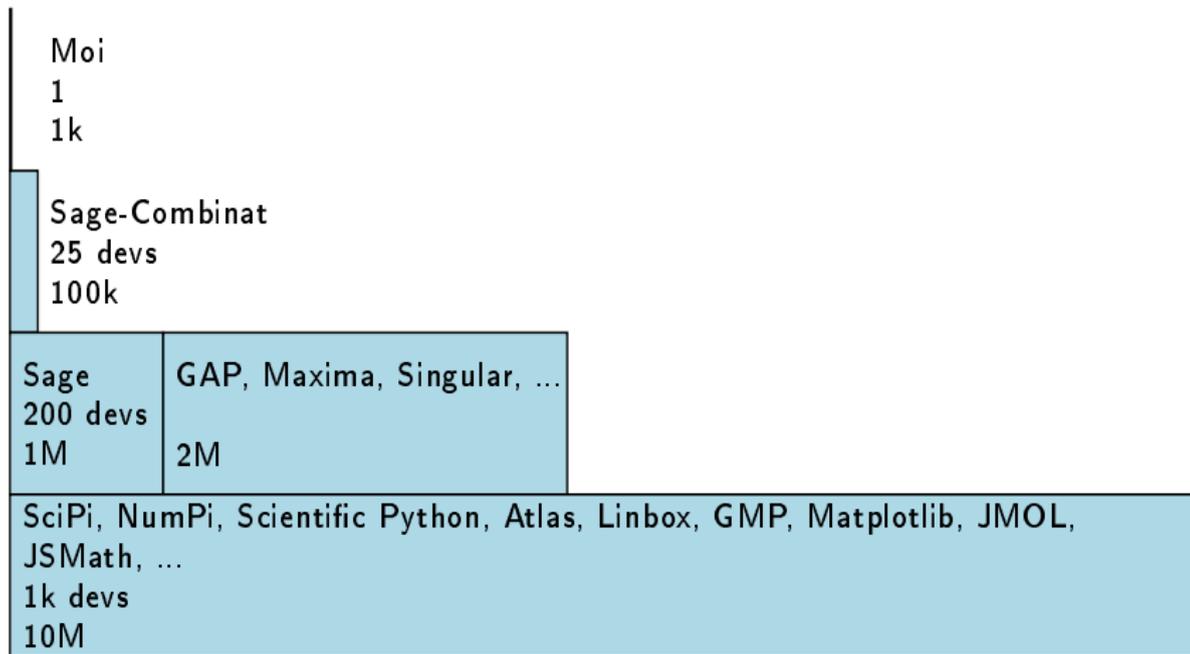
\*-Combinat

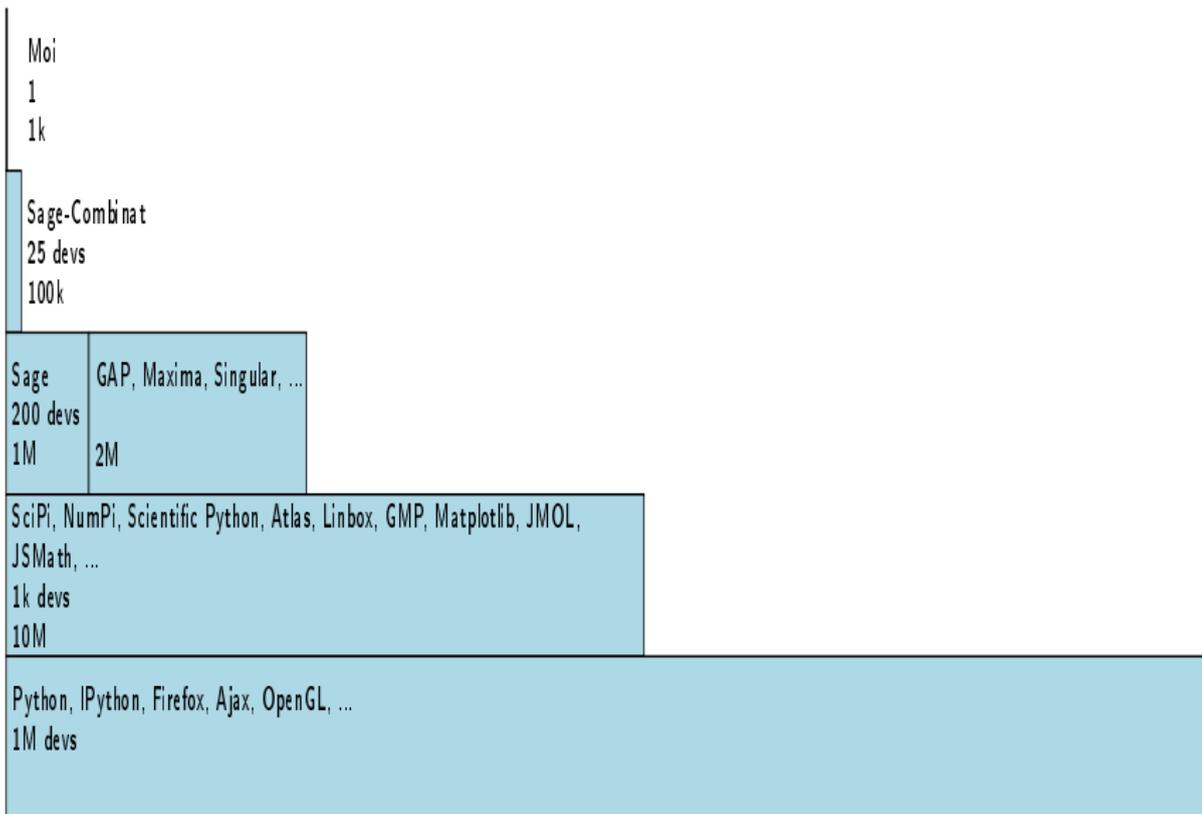
15 devs

100k

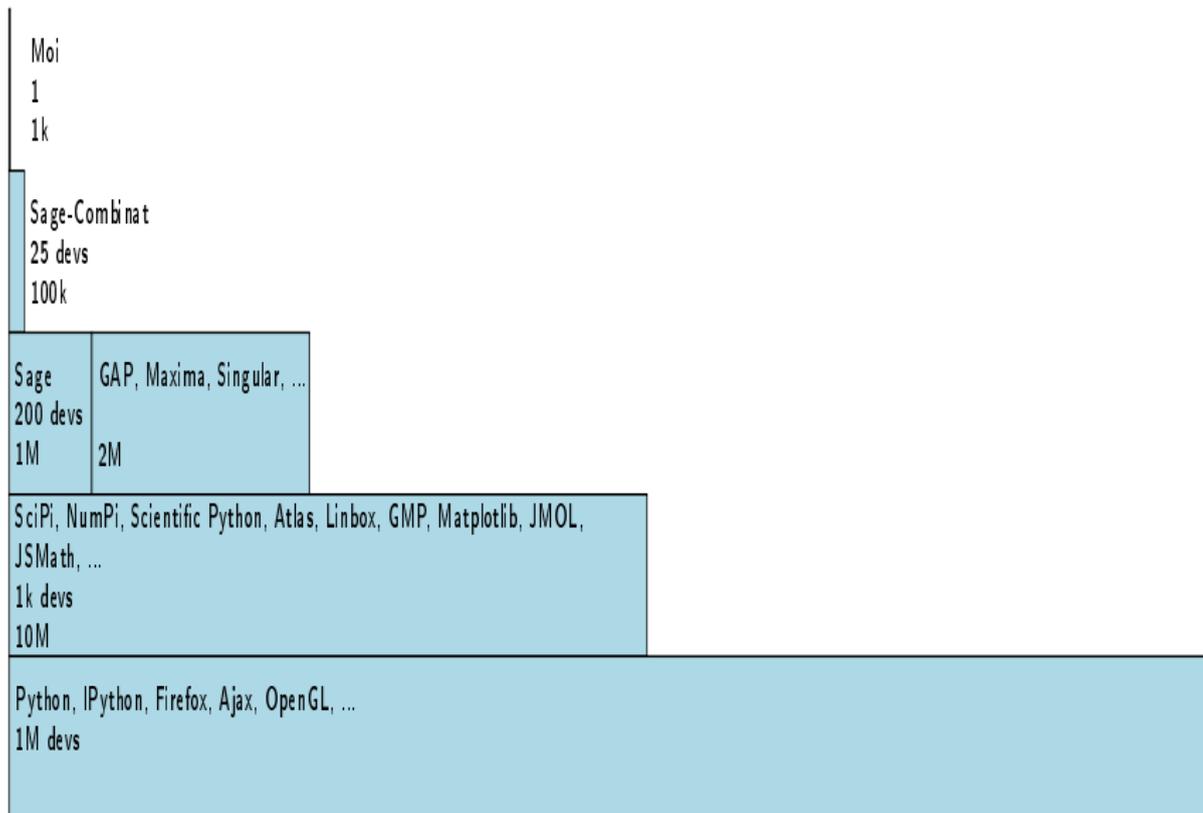


Moi 1 1k	
Sage-Combinat 25 devs 100k	
Sage 200 devs 1M	GAP, Maxima, Singular, ... 2M





# Sur les épaules d'un géant



# Évaluation

Code développé par les utilisateurs

# Évaluation

## Code développé par les utilisateurs

- Pour un besoin précis (recherche / enseignement)
- Valorisé / financé indirectement

# Évaluation

## Code développé par les utilisateurs

- Pour un besoin précis (recherche / enseignement)
- Valorisé / financé indirectement
- **Gouvernance par la demande**

# Évaluation

## Code développé par les utilisateurs

- Pour un besoin précis (recherche / enseignement)
- Valorisé / financé indirectement
- **Gouvernance par la demande**

⇒ Évaluation de l'utilité

⇒ Contribue à l'efficacité et la pérennité du développement

# Évaluation

## Code développé par les utilisateurs

- Pour un besoin précis (recherche / enseignement)
- Valorisé / financé indirectement
- **Gouvernance par la demande**

⇒ Évaluation de l'utilité

⇒ Contribue à l'efficacité et la pérennité du développement

## Revu par les pairs

# Évaluation

## Code développé par les utilisateurs

- Pour un besoin précis (recherche / enseignement)
- Valorisé / financé indirectement
- **Gouvernance par la demande**

⇒ Évaluation de l'utilité

⇒ Contribue à l'efficacité et la pérennité du développement

## Revu par les pairs

- Revue technique :
  - Généricité, modularité, réutilisation
  - Tests rigoureux
  - Bonne documentation (manuels de références, tutoriels,...)

# Évaluation

## Code développé par les utilisateurs

- Pour un besoin précis (recherche / enseignement)
- Valorisé / financé indirectement
- **Gouvernance par la demande**

⇒ Évaluation de l'utilité

⇒ Contribue à l'efficacité et la pérennité du développement

## Revu par les pairs

- Revue technique :
  - Généricité, modularité, réutilisation
  - Tests rigoureux
  - Bonne documentation (manuels de références, tutoriels,...)
- Revue mathématique

# Évaluation

## Code développé par les utilisateurs

- Pour un besoin précis (recherche / enseignement)
- Valorisé / financé indirectement
- **Gouvernance par la demande**

⇒ Évaluation de l'utilité

⇒ Contribue à l'efficacité et la pérennité du développement

## Revu par les pairs

- Revue technique :
  - Généricité, modularité, réutilisation
  - Tests rigoureux
  - Bonne documentation (manuels de références, tutoriels,...)
- Revue mathématique

⇒ Évaluation de la qualité

# Dépôt

## Outils collaboratifs (forges, ...)

- Mercurial, gestion de patchs
- Trac (gestion de tickets)
- Listes de diffusion, wiki, ...

## Dépôt

### Outils collaboratifs (forges, ...)

- Mercurial, gestion de patchs
- Trac (gestion de tickets)
- Listes de diffusion, wiki, ...

### Pourquoi contribuer au pot commun ?

- Cela prend du temps ! Et mes secrets métiers ?

## Dépôt

### Outils collaboratifs (forges, ...)

- Mercurial, gestion de patches
- Trac (gestion de tickets)
- Listes de diffusion, wiki, ...

### Pourquoi contribuer au pot commun ?

- Cela prend du temps ! Et mes secrets métiers ?
- Se forcer à écrire du code propre et générique
- Bénéficier d'une expertise externe
- Pérennité de son propre code
- Récompense sociale (ou plus)

## Dépôt

### Outils collaboratifs (forges, ...)

- Mercurial, gestion de patches
- Trac (gestion de tickets)
- Listes de diffusion, wiki, ...

### Pourquoi contribuer au pot commun ?

- Cela prend du temps ! Et mes secrets métiers ?
- Se forcer à écrire du code propre et générique
- Bénéficier d'une expertise externe
- Pérennité de son propre code
- Récompense sociale (ou plus)

⇒ Participer à la gouvernance !

## Trois clefs de la viabilité

Généricité



Évaluation

Dépôt

# Développement collaboratif d'une application métier

## Quelques prérequis

- Une communauté d'utilisateurs potentiels ?
- Une frustration sur les outils existants ?
- Une collection d'échecs ?
- Des compétences informatique dans la communauté ?
- Des financements directs ou indirects ?
- Des composants libres ? Une plateforme libre ?

# Développement collaboratif d'une application métier

## Quelques prérequis

- Une communauté d'utilisateurs potentiels ?
- Une frustration sur les outils existants ?
- Une collection d'échecs ?
- Des compétences informatique dans la communauté ?
- Des financements directs ou indirects ?
- Des composants libres ? Une plateforme libre ?

## Pour démarrer

- Un porteur de projet et un noyau dur
- Une stratégie et un modèle de développement
- Une visée décentralisée et internationale

## Pour aller plus loin

- *La cathédrale et le Bazar*, Éric Raymond
- *Économie du logiciel libre*, François Élie