



Technische
Universität
Braunschweig

Benefits of open source development and open tools

Max Horn, August 31, 2011

Overview

- My background
- What does “open source” mean, anyway?
- Why should you care?
- How to be open and make the best out of it?
- Tools, tools, tools

Overview

- **My background**
- What does “open source” mean, anyway?
- Why should you care?
- How to be open and make the best out of it?
- Tools, tools, tools

Who am I

- Math PhD 2009, “Involutions of Kac-Moody groups”
- Currently working in computer algebra, esp. GAP
- Writing code for 20+ years, primarily for Mac and Unix
- Involved in open source since 1999
- Contributed (in small and big ways) to many open source projects, e.g. ScummVM, Fink, TeXShop, SDL, SourceForge.net, XMPP chat, . . .
- Also worked as IT freelancer in the industry for a few years

Who am I

- Math PhD 2009, “Involutions of Kac-Moody groups”
- Currently working in computer algebra, esp. GAP
- Writing code for 20+ years, primarily for Mac and Unix
- Involved in open source since 1999
- Contributed (in small and big ways) to many open source projects, e.g. ScummVM, Fink, TeXShop, SDL, SourceForge.net, XMPP chat, . . .
- Also worked as IT freelancer in the industry for a few years

Who am I

- Math PhD 2009, “Involutions of Kac-Moody groups”
- Currently working in computer algebra, esp. GAP
- Writing code for 20+ years, primarily for Mac and Unix
- Involved in open source since 1999
- Contributed (in small and big ways) to many open source projects, e.g. ScummVM, Fink, TeXShop, SDL, SourceForge.net, XMPP chat, . . .
- Also worked as IT freelancer in the industry for a few years

Who am I

- Math PhD 2009, “Involutions of Kac-Moody groups”
- Currently working in computer algebra, esp. GAP
- Writing code for 20+ years, primarily for Mac and Unix
- Involved in open source since 1999
- Contributed (in small and big ways) to many open source projects, e.g. ScummVM, Fink, TeXShop, SDL, SourceForge.net, XMPP chat, . . .
- Also worked as IT freelancer in the industry for a few years

Who am I

- Math PhD 2009, “Involutions of Kac-Moody groups”
- Currently working in computer algebra, esp. GAP
- Writing code for 20+ years, primarily for Mac and Unix
- Involved in open source since 1999
- Contributed (in small and big ways) to many open source projects, e.g. ScummVM, Fink, TeXShop, SDL, SourceForge.net, XMPP chat, . . .
- Also worked as IT freelancer in the industry for a few years

Who am I

- Math PhD 2009, “Involutions of Kac-Moody groups”
- Currently working in computer algebra, esp. GAP
- Writing code for 20+ years, primarily for Mac and Unix
- Involved in open source since 1999
- Contributed (in small and big ways) to many open source projects, e.g. ScummVM, Fink, TeXShop, SDL, SourceForge.net, XMPP chat, . . .
- Also worked as IT freelancer in the industry for a few years

Example: ScummVM

- Project started in 2001, project (co-)lead 2002–2011
 - About 40 active developers in 2011; over 200 all-time
 - ~1.3M lines of code (mostly C++)
-
- ~1.2M downloads in the past twelve months
 - Highly portable: Unix, OS X, Windows, phones, game consoles, . . .

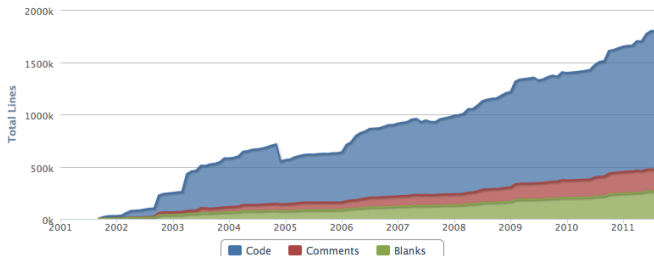
Example: ScummVM

- Project started in 2001, project (co-)lead 2002–2011
- About 40 active developers in 2011; over 200 all-time
- ~1.3M lines of code (mostly C++)

- ~1.2M downloads in the past twelve months
- Highly portable: Unix, OS X, Windows, phones, game consoles, . . .

Example: ScummVM

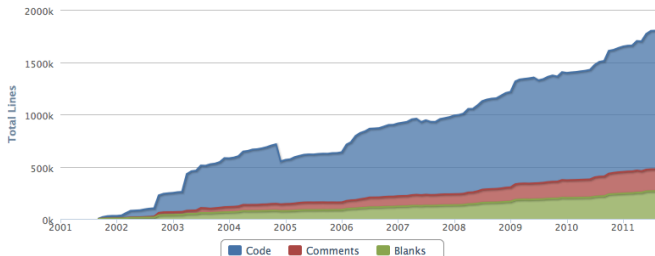
- Project started in 2001, project (co-)lead 2002–2011
- About 40 active developers in 2011; over 200 all-time
- $\sim 1.3\text{M}$ lines of code (mostly C++)



- $\sim 1.2\text{M}$ downloads in the past twelve months
- Highly portable: Unix, OS X, Windows, phones, game consoles, . . .

Example: ScummVM

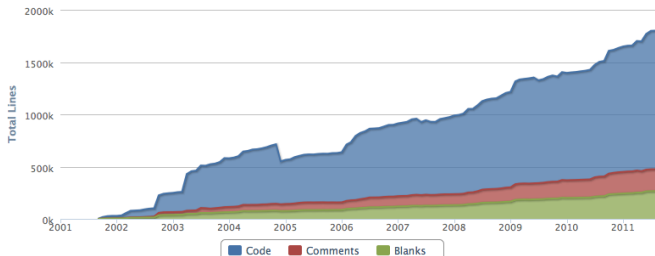
- Project started in 2001, project (co-)lead 2002–2011
- About 40 active developers in 2011; over 200 all-time
- $\sim 1.3\text{M}$ lines of code (mostly C++)



- $\sim 1.2\text{M}$ downloads in the past twelve months
- Highly portable: Unix, OS X, Windows, phones, game consoles, . . .

Example: ScummVM

- Project started in 2001, project (co-)lead 2002–2011
- About 40 active developers in 2011; over 200 all-time
- $\sim 1.3\text{M}$ lines of code (mostly C++)



- $\sim 1.2\text{M}$ downloads in the past twelve months
- Highly portable: Unix, OS X, Windows, phones, game consoles, . . .

Overview

- My background
- **What does “open source” mean, anyway?**
- Why should you care?
- How to be open and make the best out of it?
- Tools, tools, tools

What does “open source” mean, anyway?

- You probably already have an idea of what **open source** means . . .
- . . . but let's make sure we are on the same page here!
- Wikipedia says:

The term open source describes practices in production and development that promote access to the end product's source materials. Some consider open source a philosophy, others consider it a pragmatic methodology.

- Applied to software, this leads to **open source software**; for example: Linux (kernel), OpenOffice, but also Sage, GAP, Singular, CoCoA, . . .
- Usually assumed to mean access to the **complete** source for **everybody**
- Various **licenses** exist that are meant to ensure this from a legal point (BSD, GNU General Public License v2 and v3, and many more)

What does “open source” mean, anyway?

- You probably already have an idea of what **open source** means . . .
- . . . but let’s make sure we are on the same page here!
- Wikipedia says:

The term open source describes practices in production and development that promote access to the end product’s source materials. Some consider open source a philosophy, others consider it a pragmatic methodology.

- Applied to software, this leads to **open source software**; for example: Linux (kernel), OpenOffice, but also Sage, GAP, Singular, CoCoA, . . .
- Usually assumed to mean access to the **complete** source for **everybody**
- Various **licenses** exist that are meant to ensure this from a legal point (BSD, GNU General Public License v2 and v3, and many more)

What does “open source” mean, anyway?

- You probably already have an idea of what **open source** means . . .
- . . . but let’s make sure we are on the same page here!
- Wikipedia says:

The term open source describes practices in production and development that promote access to the end product’s source materials. Some consider open source a philosophy, others consider it a pragmatic methodology.

- Applied to software, this leads to **open source software**; for example: Linux (kernel), OpenOffice, but also Sage, GAP, Singular, CoCoA, . . .
- Usually assumed to mean access to the **complete** source for **everybody**
- Various **licenses** exist that are meant to ensure this from a legal point (BSD, GNU General Public License v2 and v3, and many more)

What does “open source” mean, anyway?

- You probably already have an idea of what **open source** means . . .
- . . . but let’s make sure we are on the same page here!
- Wikipedia says:

The term open source describes practices in production and development that promote access to the end product’s source materials. Some consider open source a philosophy, others consider it a pragmatic methodology.

- Applied to software, this leads to **open source software**; for example: Linux (kernel), OpenOffice, but also Sage, GAP, Singular, CoCoA, . . .
- Usually assumed to mean access to the **complete** source for **everybody**
- Various **licenses** exist that are meant to ensure this from a legal point (BSD, GNU General Public License v2 and v3, and many more)

What does “open source” mean, anyway?

- You probably already have an idea of what **open source** means . . .
- . . . but let’s make sure we are on the same page here!
- Wikipedia says:

The term open source describes practices in production and development that promote access to the end product’s source materials. Some consider open source a philosophy, others consider it a pragmatic methodology.

- Applied to software, this leads to **open source software**; for example: Linux (kernel), OpenOffice, but also Sage, GAP, Singular, CoCoA, . . .
- Usually assumed to mean access to the **complete** source for **everybody**
- Various **licenses** exist that are meant to ensure this from a legal point (BSD, GNU General Public License v2 and v3, and many more)

What does “open source” mean, anyway?

- You probably already have an idea of what **open source** means . . .
- . . . but let’s make sure we are on the same page here!
- Wikipedia says:

The term open source describes practices in production and development that promote access to the end product’s source materials. Some consider open source a philosophy, others consider it a pragmatic methodology.

- Applied to software, this leads to **open source software**; for example: Linux (kernel), OpenOffice, but also Sage, GAP, Singular, CoCoA, . . .
- Usually assumed to mean access to the **complete** source for **everybody**
- Various **licenses** exist that are meant to ensure this from a legal point (BSD, GNU General Public License v2 and v3, and many more)

What does “open source” mean, anyway?

- You probably already have an idea of what **open source** means . . .
- . . . but let’s make sure we are on the same page here!
- Wikipedia says:

The term open source describes practices in production and development that promote access to the end product’s source materials. Some consider open source a philosophy, others consider it a pragmatic methodology.

- Applied to software, this leads to **open source software**; for example: Linux (kernel), OpenOffice, but also Sage, GAP, Singular, CoCoA, . . .
- Usually assumed to mean access to the **complete** source for **everybody**
- Various **licenses** exist that are meant to ensure this from a legal point (BSD, GNU General Public License v2 and v3, and many more)

Open source goes beyond software

- Most people associated “open source” with software alone ...
- ... but there is much more to it.
- E.g. open source hardware (OpenSPARC T1 processor, Arduino, ...)

Open source goes beyond software

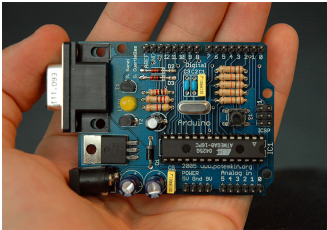
- Most people associated “open source” with software alone ...
- ... but there is much more to it.
- E.g. open source hardware (OpenSPARC T1 processor, Arduino, ...)

Open source goes beyond software

- Most people associated “open source” with software alone ...
- ... but there is much more to it.
- E.g. open source hardware (OpenSPARC T1 processor, Arduino, ...)

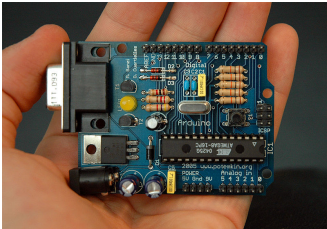
Open source goes beyond software

- Most people associated “open source” with software alone ...
- ... but there is much more to it.
- E.g. open source hardware (OpenSPARC T1 processor, Arduino, ...)



Open source goes beyond software

- Most people associated “open source” with software alone ...
- ... but there is much more to it.
- E.g. open source hardware (OpenSPARC T1 processor, Arduino, ...)



... and beverages! ☺



Publications = open source?

- Research publications (in math) are similar to open source:
 - The theorems in a paper are the **product**.
 - The proofs and lemmas are the **source**, and they are “shipped” along with the product.
- Being open allows the **community** to advance much quicker:
- “Standing on the shoulders of giants”
- Now imagine we would only ship the “product”, not the “source”!
- For individuals, this can be tempting, e.g. to hinder competition ...
- ... but for the community it is a big loss.

Publications = open source?

- Research publications (in math) are similar to open source:
 - The theorems in a paper are the **product**.
 - The proofs and lemmas are the **source**, and they are “shipped” along with the product.
- Being open allows the **community** to advance much quicker:
- “Standing on the shoulders of giants”
- Now imagine we would only ship the “product”, not the “source”!
- For individuals, this can be tempting, e.g. to hinder competition . . .
- . . . but for the community it is a big loss.

Publications = open source?

- Research publications (in math) are similar to open source:
 - The theorems in a paper are the **product**.
 - The proofs and lemmas are the **source**, and they are “shipped” along with the product.
- Being open allows the **community** to advance much quicker:
- “Standing on the shoulders of giants”
- Now imagine we would only ship the “product”, not the “source”!
- For individuals, this can be tempting, e.g. to hinder competition ...
- ... but for the community it is a big loss.

Publications = open source?

- Research publications (in math) are similar to open source:
 - The theorems in a paper are the **product**.
 - The proofs and lemmas are the **source**, and they are “shipped” along with the product.
- Being open allows the **community** to advance much quicker:
 - “Standing on the shoulders of giants”
 - Now imagine we would only ship the “product”, not the “source”!
 - For individuals, this can be tempting, e.g. to hinder competition ...
 - ... but for the community it is a big loss.

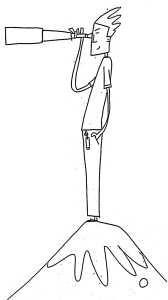
Publications = open source?

- Research publications (in math) are similar to open source:
 - The theorems in a paper are the **product**.
 - The proofs and lemmas are the **source**, and they are “shipped” along with the product.
- Being open allows the **community** to advance much quicker:
- “Standing on the shoulders of giants”
 - Now imagine we would only ship the “product”, not the “source”!
 - For individuals, this can be tempting, e.g. to hinder competition ...
 - ... but for the community it is a big loss.



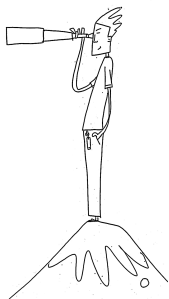
Publications = open source?

- Research publications (in math) are similar to open source:
 - The theorems in a paper are the **product**.
 - The proofs and lemmas are the **source**, and they are “shipped” along with the product.
- Being open allows the **community** to advance much quicker:
- “Standing on the shoulders of giants”
- Now imagine we would only ship the “product”, not the “source”!
- For individuals, this can be tempting, e.g. to hinder competition ...
- ... but for the community it is a big loss.



Publications = open source?

- Research publications (in math) are similar to open source:
 - The theorems in a paper are the **product**.
 - The proofs and lemmas are the **source**, and they are “shipped” along with the product.
- Being open allows the **community** to advance much quicker:
- “Standing on the shoulders of giants”
- Now imagine we would only ship the “product”, not the “source”!
- For individuals, this can be tempting, e.g. to hinder competition ...
- ... but for the community it is a big loss.



Publications = open source?

- Research publications (in math) are similar to open source:
 - The theorems in a paper are the **product**.
 - The proofs and lemmas are the **source**, and they are “shipped” along with the product.
- Being open allows the **community** to advance much quicker:
- “Standing on the shoulders of giants”
- Now imagine we would only ship the “product”, not the “source”!
- For individuals, this can be tempting, e.g. to hinder competition ...
- ... but for the community it is a big loss.



Overview

- My background
- What does “open source” mean, anyway?
- **Why should you care?**
- How to be open and make the best out of it?
- Tools, tools, tools

Benefits of open source

We saw what is good about “open source publications”. For software, it is similar:

- Releasing your code allows others to learn from it
- ... and sometimes to improve on it
- ... and to build upon it.
- Again: “Standing on the shoulders of giants”
- Trust: No software is bug-free. No source = no idea of what is going on.
- Allows code to live on even after author lost interest in (or time for) it.

Benefits of open source

We saw what is good about “open source publications”. For software, it is similar:

- Releasing your code allows others to learn from it
- ... and sometimes to improve on it
- ... and to build upon it.
- Again: “Standing on the shoulders of giants”
- Trust: No software is bug-free. No source = no idea of what is going on.
- Allows code to live on even after author lost interest in (or time for) it.

Benefits of open source

We saw what is good about “open source publications”. For software, it is similar:

- Releasing your code allows others to learn from it
- ... and sometimes to improve on it
- ... and to build upon it.
- Again: “Standing on the shoulders of giants”
- Trust: No software is bug-free. No source = no idea of what is going on.
- Allows code to live on even after author lost interest in (or time for) it.

Benefits of open source

We saw what is good about “open source publications”. For software, it is similar:

- Releasing your code allows others to learn from it
- ... and sometimes to improve on it
- ... and to build upon it.
- Again: “Standing on the shoulders of giants”
- Trust: No software is bug-free. No source = no idea of what is going on.
- Allows code to live on even after author lost interest in (or time for) it.

Benefits of open source

We saw what is good about “open source publications”. For software, it is similar:

- Releasing your code allows others to learn from it
- ... and sometimes to improve on it
- ... and to build upon it.
- Again: “Standing on the shoulders of giants”
- Trust: No software is bug-free. No source = no idea of what is going on.
- Allows code to live on even after author lost interest in (or time for) it.

Benefits of open source

We saw what is good about “open source publications”. For software, it is similar:

- Releasing your code allows others to learn from it
- ... and sometimes to improve on it
- ... and to build upon it.
- Again: “Standing on the shoulders of giants”
- Trust: No software is bug-free. No source = no idea of what is going on.
- Allows code to live on even after author lost interest in (or time for) it.

Benefits of open source

We saw what is good about “open source publications”. For software, it is similar:

- Releasing your code allows others to learn from it
- ... and sometimes to improve on it
- ... and to build upon it.
- Again: “Standing on the shoulders of giants”
- Trust: No software is bug-free. No source = no idea of what is going on.
- Allows code to live on even after author lost interest in (or time for) it.
 - Alnuth 2.x (GAP package) needs KASH 2.5

Benefits of open source

We saw what is good about “open source publications”. For software, it is similar:

- Releasing your code allows others to learn from it
- ... and sometimes to improve on it
- ... and to build upon it.
- Again: “Standing on the shoulders of giants”
- Trust: No software is bug-free. No source = no idea of what is going on.
- Allows code to live on even after author lost interest in (or time for) it.
 - Alnuth 2.x (GAP package) needs KASH 2.5

Benefits of open source

We saw what is good about “open source publications”. For software, it is similar:

- Releasing your code allows others to learn from it
- ... and sometimes to improve on it
- ... and to build upon it.
- Again: “Standing on the shoulders of giants”
- Trust: No software is bug-free. No source = no idea of what is going on.
- Allows code to live on even after author lost interest in (or time for) it.
 - Alnuth 2.x (GAP package) needs KASH 2.5

This version has been ported to the following architectures:

- AIX - PPC/RS6000: [kash_2.5.1.aix.tar.gz](#)
- Linux - x86: [kash_2.5.7.linux.tar.gz](#)
- Mac OS X 10.3 - PPC: [kash_2.5.6.macosex.tar.gz](#)
- Solaris 2 - SPARC: [kash_2.5.4.solaris.tar.gz](#)

Benefits of open source

We saw what is good about “open source publications”. For software, it is similar:

- Releasing your code allows others to learn from it
- ... and sometimes to improve on it
- ... and to build upon it.
- Again: “Standing on the shoulders of giants”
- Trust: No software is bug-free. No source = no idea of what is going on.
- Allows code to live on even after author lost interest in (or time for) it.
 - Alnuth 2.x (GAP package) needs KASH 2.5
 - Alnuth 3.0 uses open source Pari/GP ☺

This version has been ported to the following architectures:

- AIX - PPC/RS6000: [kash_2.5.1.aix.tar.gz](#)
- Linux - x86: [kash_2.5.7.linux.tar.gz](#)
- Mac OS X 10.3 - PPC: [kash_2.5.6.macosex.tar.gz](#)
- Solaris 2 - SPARC: [kash_2.5.4.solaris.tar.gz](#)

The dark side

- *“But what if my code gets stolen?”*
 - The possibility exists, no denying it . . .
 - . . . but not more so than for research ideas and papers.
 - Same countermeasures generally work well:
 - Legal protection via copyright (remember those licenses?);
 - peer review.
 - Example: ScummVM (GPL v2) illegally sold as part of software for Nintendo Wii. . .
- *“But my code is so ugly, I don't want to show it, and nobody will be able to read it anyway!”*
 - *“But my proof is so ugly, I don't want to publish it.”*
 - Your code might not be as bad as you think.
 - Maybe you should improve it then, anyway?
 - Maybe others will help improving it if your tool is useful to them?

The dark side

- *“But what if my code gets stolen?”*
 - The possibility exists, no denying it . . .
 - . . . but not more so than for research ideas and papers.
 - Same countermeasures generally work well:
 - Legal protection via copyright (remember those licenses?);
 - peer review.
 - Example: ScummVM (GPL v2) illegally sold as part of software for Nintendo Wii. . .
- *“But my code is so ugly, I don't want to show it, and nobody will be able to read it anyway!”*
 - *“But my proof is so ugly, I don't want to publish it.”*
 - Your code might not be as bad as you think.
 - Maybe you should improve it then, anyway?
 - Maybe others will help improving it if your tool is useful to them?

The dark side

- *“But what if my code gets stolen?”*
 - The possibility exists, no denying it . . .
 - . . . but not more so than for research ideas and papers.
 - Same countermeasures generally work well:
 - Legal protection via copyright (remember those licenses?);
 - peer review.
 - Example: ScummVM (GPL v2) illegally sold as part of software for Nintendo Wii. . .
- *“But my code is so ugly, I don't want to show it, and nobody will be able to read it anyway!”*
 - *“But my proof is so ugly, I don't want to publish it.”*
 - Your code might not be as bad as you think.
 - Maybe you should improve it then, anyway?
 - Maybe others will help improving it if your tool is useful to them?

The dark side

- *“But what if my code gets stolen?”*
 - The possibility exists, no denying it . . .
 - . . . but not more so than for research ideas and papers.
 - Same countermeasures generally work well:
 - Legal protection via copyright (remember those licenses?);
 - peer review.
 - Example: ScummVM (GPL v2) illegally sold as part of software for Nintendo Wii. . .
- *“But my code is so ugly, I don't want to show it, and nobody will be able to read it anyway!”*
 - *“But my proof is so ugly, I don't want to publish it.”*
 - Your code might not be as bad as you think.
 - Maybe you should improve it then, anyway?
 - Maybe others will help improving it if your tool is useful to them?

The dark side

- *“But what if my code gets stolen?”*
 - The possibility exists, no denying it . . .
 - . . . but not more so than for research ideas and papers.
 - Same countermeasures generally work well:
 - Legal protection via copyright (remember those licenses?);
 - peer review.
 - Example: ScummVM (GPL v2) illegally sold as part of software for Nintendo Wii. . .
- *“But my code is so ugly, I don't want to show it, and nobody will be able to read it anyway!”*
 - *“But my proof is so ugly, I don't want to publish it.”*
 - Your code might not be as bad as you think.
 - Maybe you should improve it then, anyway?
 - Maybe others will help improving it if your tool is useful to them?

The dark side

- *“But what if my code gets stolen?”*
 - The possibility exists, no denying it . . .
 - . . . but not more so than for research ideas and papers.
 - Same countermeasures generally work well:
 - Legal protection via copyright (remember those licenses?);
 - peer review.
 - Example: ScummVM (GPL v2) illegally sold as part of software for Nintendo Wii. . .
- *“But my code is so ugly, I don't want to show it, and nobody will be able to read it anyway!”*
 - *“But my proof is so ugly, I don't want to publish it.”*
 - Your code might not be as bad as you think.
 - Maybe you should improve it then, anyway?
 - Maybe others will help improving it if your tool is useful to them?

The dark side

- *“But what if my code gets stolen?”*
 - The possibility exists, no denying it . . .
 - . . . but not more so than for research ideas and papers.
 - Same countermeasures generally work well:
 - Legal protection via copyright (remember those licenses?);
 - peer review.
 - Example: ScummVM (GPL v2) illegally sold as part of software for Nintendo Wii. . .
- *“But my code is so ugly, I don't want to show it, and nobody will be able to read it anyway!”*
 - *“But my proof is so ugly, I don't want to publish it.”*
 - Your code might not be as bad as you think.
 - Maybe you should improve it then, anyway?
 - Maybe others will help improving it if your tool is useful to them?

The dark side

- *“But what if my code gets stolen?”*
 - The possibility exists, no denying it . . .
 - . . . but not more so than for research ideas and papers.
 - Same countermeasures generally work well:
 - Legal protection via copyright (remember those licenses?);
 - peer review.
 - Example: ScummVM (GPL v2) illegally sold as part of software for Nintendo Wii. . .
- *“But my code is so ugly, I don’t want to show it, and nobody will be able to read it anyway!”*
 - *“But my proof is so ugly, I don’t want to publish it.”*
 - Your code might not be as bad as you think.
 - Maybe you should improve it then, anyway?
 - Maybe others will help improving it if your tool is useful to them?

The dark side

- *“But what if my code gets stolen?”*
 - The possibility exists, no denying it . . .
 - . . . but not more so than for research ideas and papers.
 - Same countermeasures generally work well:
 - Legal protection via copyright (remember those licenses?);
 - peer review.
 - Example: ScummVM (GPL v2) illegally sold as part of software for Nintendo Wii. . .
- *“But my code is so ugly, I don’t want to show it, and nobody will be able to read it anyway!”*
 - *“But my proof is so ugly, I don’t want to publish it.”*
 - Your code might not be as bad as you think.
 - Maybe you should improve it then, anyway?
 - Maybe others will help improving it if your tool is useful to them?

The dark side

- *“But what if my code gets stolen?”*
 - The possibility exists, no denying it . . .
 - . . . but not more so than for research ideas and papers.
 - Same countermeasures generally work well:
 - Legal protection via copyright (remember those licenses?);
 - peer review.
 - Example: ScummVM (GPL v2) illegally sold as part of software for Nintendo Wii. . .
- *“But my code is so ugly, I don’t want to show it, and nobody will be able to read it anyway!”*
 - *“But my proof is so ugly, I don’t want to publish it.”*
 - Your code might not be as bad as you think.
 - Maybe you should improve it then, anyway?
 - Maybe others will help improving it if your tool is useful to them?

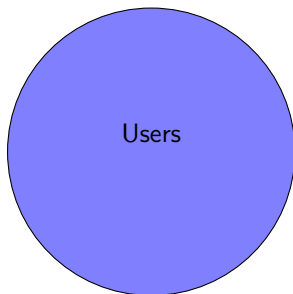
The dark side

- *“But what if my code gets stolen?”*
 - The possibility exists, no denying it . . .
 - . . . but not more so than for research ideas and papers.
 - Same countermeasures generally work well:
 - Legal protection via copyright (remember those licenses?);
 - peer review.
 - Example: ScummVM (GPL v2) illegally sold as part of software for Nintendo Wii. . .
- *“But my code is so ugly, I don’t want to show it, and nobody will be able to read it anyway!”*
 - *“But my proof is so ugly, I don’t want to publish it.”*
 - Your code might not be as bad as you think.
 - Maybe you should improve it then, anyway?
 - Maybe others will help improving it if your tool is useful to them?

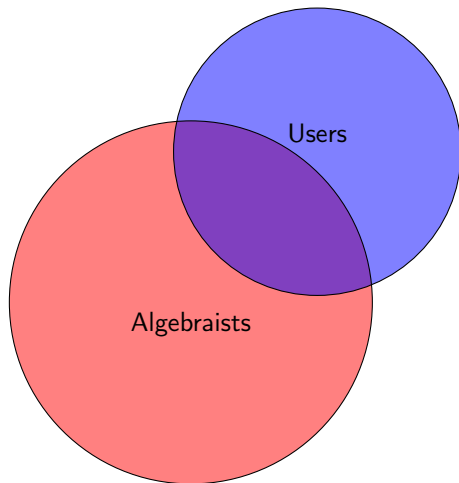
The dark side

- *“But what if my code gets stolen?”*
 - The possibility exists, no denying it . . .
 - . . . but not more so than for research ideas and papers.
 - Same countermeasures generally work well:
 - Legal protection via copyright (remember those licenses?);
 - peer review.
 - Example: ScummVM (GPL v2) illegally sold as part of software for Nintendo Wii. . .
- *“But my code is so ugly, I don’t want to show it, and nobody will be able to read it anyway!”*
 - *“But my proof is so ugly, I don’t want to publish it.”*
 - Your code might not be as bad as you think.
 - Maybe you should improve it then, anyway?
 - Maybe others will help improving it if your tool is useful to them?

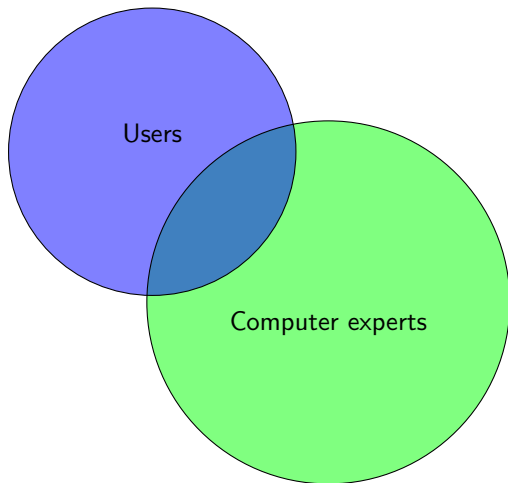
Users are a valuable resource



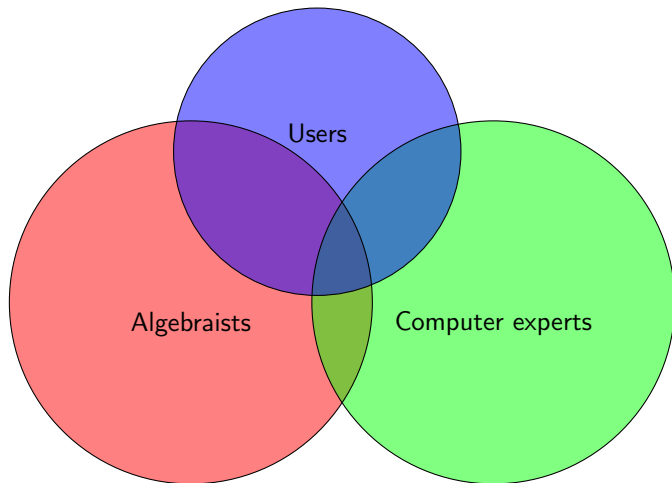
Users are a valuable resource



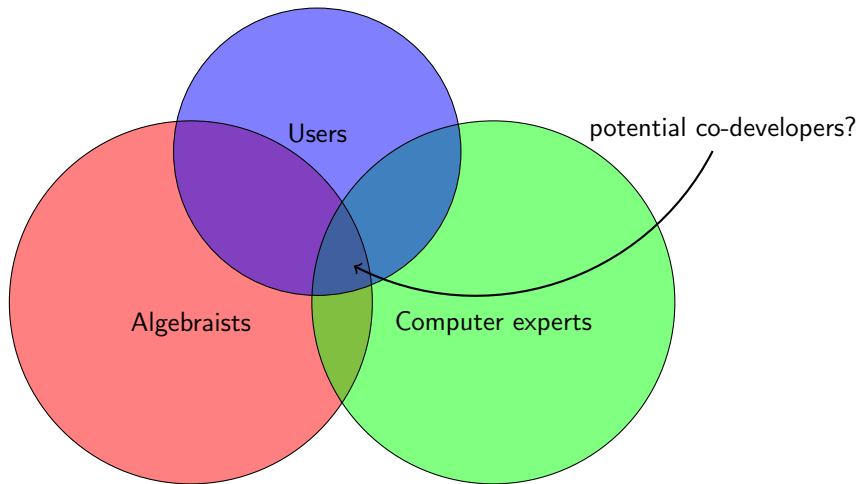
Users are a valuable resource



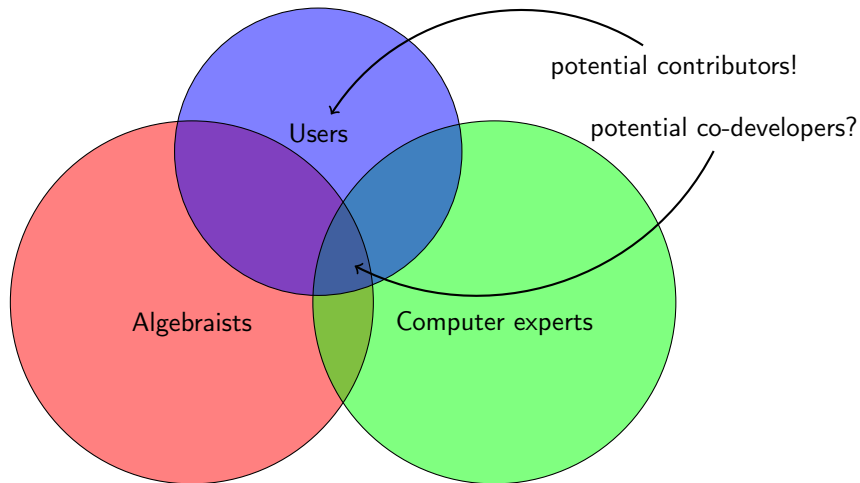
Users are a valuable resource



Users are a valuable resource



Users are a valuable resource



Beyond open source: Open development

- We saw the technical definition of open source. . .
- . . . but for many, it means much more:
- An **open development process!**
- This means for example:
 - everybody can see the current code at all times, not just for releases;
 - discussions and planning is done in the open, everybody can chime in;
 - everybody can submit code patches.
- Does **not** mean everybody gets the same say; your past track record is import (“meritocracy”).

Beyond open source: Open development

- We saw the technical definition of open source. . .
- . . . but for many, it means much more:
- An **open development process!**
- This means for example:
 - everybody can see the current code at all times, not just for releases;
 - discussions and planning is done in the open, everybody can chime in;
 - everybody can submit code patches.
- Does **not** mean everybody gets the same say; your past track record is import (“meritocracy”).

Beyond open source: Open development

- We saw the technical definition of open source. . .
- . . . but for many, it means much more:
- An **open development process!**
- This means for example:
 - everybody can see the current code at all times, not just for releases;
 - discussions and planning is done in the open, everybody can chime in;
 - everybody can submit code patches.
- Does **not** mean everybody gets the same say; your past track record is import (“meritocracy”).

Beyond open source: Open development

- We saw the technical definition of open source. . .
- . . . but for many, it means much more:
- An **open development process!**
- This means for example:
 - everybody can see the current code at all times, not just for releases;
 - discussions and planning is done in the open, everybody can chime in;
 - everybody can submit code patches.
- Does **not** mean everybody gets the same say; your past track record is import (“meritocracy”).

Beyond open source: Open development

- We saw the technical definition of open source. . .
- . . . but for many, it means much more:
- An **open development process!**
- This means for example:
 - everybody can see the current code at all times, not just for releases;
 - discussions and planning is done in the open, everybody can chime in;
 - everybody can submit code patches.
- Does **not** mean everybody gets the same say; your past track record is import (“meritocracy”).

Beyond open source: Open development

- We saw the technical definition of open source. . .
- . . . but for many, it means much more:
- An **open development process!**
- This means for example:
 - everybody can see the current code at all times, not just for releases;
 - discussions and planning is done in the open, everybody can chime in;
 - everybody can submit code patches.
- Does **not** mean everybody gets the same say; your past track record is import (“meritocracy”).

Beyond open source: Open development

- We saw the technical definition of open source. . .
- . . . but for many, it means much more:
- An **open development process!**
- This means for example:
 - everybody can see the current code at all times, not just for releases;
 - discussions and planning is done in the open, everybody can chime in;
 - everybody can submit code patches.
- Does **not** mean everybody gets the same say; your past track record is import (“meritocracy”).

Beyond open source: Open development

- We saw the technical definition of open source. . .
- . . . but for many, it means much more:
- An **open development process!**
- This means for example:
 - everybody can see the current code at all times, not just for releases;
 - discussions and planning is done in the open, everybody can chime in;
 - everybody can submit code patches.
- Does **not** mean everybody gets the same say; your past track record is import (“meritocracy”).

The Cathedral and the Bazaar

- “The Cathedral and the Bazaar” an essay by Eric S. Raymond on software engineering methods.
- **The Cathedral model:** Top-down design; source code is available with each software release, but access to code developed between releases is restricted. (Past) examples: GNU Emacs, GCC, GAP, ...
- **The Bazaar model:** Bottom-up design; code is developed over the Internet in view of the public. Examples: Linux, Sage, ScummVM, ...
- “Release early. Release often. And listen to your customers.”
- Linus’ Law: “Given enough eyeballs, all bugs are shallow.”
- Both approaches have pros and cons. The trick is to find the right mix!

The Cathedral and the Bazaar

- “The Cathedral and the Bazaar” an essay by Eric S. Raymond on software engineering methods.
- **The Cathedral model:** Top-down design; source code is available with each software release, but access to code developed between releases is restricted. (Past) examples: GNU Emacs, GCC, GAP, ...
- **The Bazaar model:** Bottom-up design; code is developed over the Internet in view of the public. Examples: Linux, Sage, ScummVM, ...
- “Release early. Release often. And listen to your customers.”
- Linus’ Law: “Given enough eyeballs, all bugs are shallow.”
- Both approaches have pros and cons. The trick is to find the right mix!

The Cathedral and the Bazaar

- “The Cathedral and the Bazaar” an essay by Eric S. Raymond on software engineering methods.
- **The Cathedral model:** Top-down design; source code is available with each software release, but access to code developed between releases is restricted. (Past) examples: ~~GNU Emacs~~, GCC, GAP, ...
- **The Bazaar model:** Bottom-up design; code is developed over the Internet in view of the public. Examples: Linux, Sage, ScummVM, ...
- “Release early. Release often. And listen to your customers.”
- Linus’ Law: “Given enough eyeballs, all bugs are shallow.”
- Both approaches have pros and cons. The trick is to find the right mix!

The Cathedral and the Bazaar

- “The Cathedral and the Bazaar” an essay by Eric S. Raymond on software engineering methods.
- **The Cathedral model:** Top-down design; source code is available with each software release, but access to code developed between releases is restricted. (Past) examples: ~~GNU Emacs~~, ~~GCC~~, GAP, ...
- **The Bazaar model:** Bottom-up design; code is developed over the Internet in view of the public. Examples: Linux, Sage, ScummVM, ...
- “Release early. Release often. And listen to your customers.”
- Linus’ Law: “Given enough eyeballs, all bugs are shallow.”
- Both approaches have pros and cons. The trick is to find the right mix!

The Cathedral and the Bazaar

- “The Cathedral and the Bazaar” an essay by Eric S. Raymond on software engineering methods.
- **The Cathedral model:** Top-down design; source code is available with each software release, but access to code developed between releases is restricted. (Past) examples: ~~GNU Emacs~~, ~~gcc~~, GAP, ...
- **The Bazaar model:** Bottom-up design; code is developed over the Internet in view of the public. Examples: Linux, Sage, ScummVM, ...
- “Release early. Release often. And listen to your customers.”
- Linus’ Law: “Given enough eyeballs, all bugs are shallow.”
- Both approaches have pros and cons. The trick is to find the right mix!

The Cathedral and the Bazaar

- “The Cathedral and the Bazaar” an essay by Eric S. Raymond on software engineering methods.
- **The Cathedral model:** Top-down design; source code is available with each software release, but access to code developed between releases is restricted. (Past) examples: ~~GNU Emacs~~, ~~gcc~~, GAP, ...
- **The Bazaar model:** Bottom-up design; code is developed over the Internet in view of the public. Examples: Linux, Sage, ScummVM, ...
- “Release early. Release often. And listen to your customers.”
- Linus’ Law: “Given enough eyeballs, all bugs are shallow.”
- Both approaches have pros and cons. The trick is to find the right mix!

The Cathedral and the Bazaar

- “The Cathedral and the Bazaar” an essay by Eric S. Raymond on software engineering methods.
- **The Cathedral model:** Top-down design; source code is available with each software release, but access to code developed between releases is restricted. (Past) examples: ~~GNU Emacs~~, ~~GCC~~, GAP, ...
- **The Bazaar model:** Bottom-up design; code is developed over the Internet in view of the public. Examples: Linux, Sage, ScummVM, ...
- “Release early. Release often. And listen to your customers.”
- Linus’ Law: “Given enough eyeballs, all bugs are shallow.”
- Both approaches have pros and cons. The trick is to find the right mix!

The Cathedral and the Bazaar

- “The Cathedral and the Bazaar” an essay by Eric S. Raymond on software engineering methods.
- **The Cathedral model:** Top-down design; source code is available with each software release, but access to code developed between releases is restricted. (Past) examples: ~~GNU Emacs~~, ~~GCC~~, GAP, ...
- **The Bazaar model:** Bottom-up design; code is developed over the Internet in view of the public. Examples: Linux, Sage, ScummVM, ...
- “Release early. Release often. And listen to your customers.”
- Linus’ Law: “Given enough eyeballs, all bugs are shallow.”
- Both approaches have pros and cons. The trick is to find the right mix!

Overview

- My background
- What does “open source” mean, anyway?
- Why should you care?
- **How to be open and make the best out of it?**
- Tools, tools, tools

Making the best out of your version control system (VCS)

- Every non-trivial software project should use a VCS repository!
- Are you ??? ☺
- Helps with understanding the code and its past (“history mining”)
- Helps tracking down regressions
- “Whom should I ask about this code?”
- What has changed since the last release?

Making the best out of your version control system (VCS)

- Every non-trivial software project should use a VCS repository!
- Are you ??? ☺
- Helps with understanding the code and its past (“history mining”)
- Helps tracking down regressions
- “Whom should I ask about this code?”
- What has changed since the last release?

Making the best out of your version control system (VCS)

- Every non-trivial software project should use a VCS repository!
- Are you ??? ☺
- Helps with understanding the code and its past (“history mining”)
- Helps tracking down regressions
- “Whom should I ask about this code?”
- What has changed since the last release?

Making the best out of your version control system (VCS)

- Every non-trivial software project should use a VCS repository!
- Are you ??? ☺
- Helps with understanding the code and its past (“history mining”)
- Helps tracking down regressions
- “Whom should I ask about this code?”
- What has changed since the last release?

Making the best out of your version control system (VCS)

- Every non-trivial software project should use a VCS repository!
- Are you ??? ☺
- Helps with understanding the code and its past (“history mining”)
- Helps tracking down regressions
- “Whom should I ask about this code?”
- What has changed since the last release?

Making the best out of your version control system (VCS)

- Every non-trivial software project should use a VCS repository!
- Are you ??? ☺
- Helps with understanding the code and its past (“history mining”)
- Helps tracking down regressions
- “Whom should I ask about this code?”
- What has changed since the last release?

... so open up your VCS!

For open development, your repository should be readable by anybody, anytime!

- Users see whether the project is still active.
- Users see where the project is heading.
- Yup, sometimes that includes dead ends ...
- ... but users can be taught about that, too.

... so open up your VCS!

For open development, your repository should be readable by anybody, anytime!

- Users see whether the project is still active.
- Users see where the project is heading.
- Yup, sometimes that includes dead ends ...
- ... but users can be taught about that, too.

... so open up your VCS!

For open development, your repository should be readable by anybody, anytime!

- Users see whether the project is still active.
- Users see where the project is heading.
- Yup, sometimes that includes dead ends ...
- ... but users can be taught about that, too.

... so open up your VCS!

For open development, your repository should be readable by anybody, anytime!

- Users see whether the project is still active.
- Users see where the project is heading.
- Yup, sometimes that includes dead ends ...
- ... but users can be taught about that, too.

... so open up your VCS!

For open development, your repository should be readable by anybody, anytime!

- Users see whether the project is still active.
- Users see where the project is heading.
- Yup, sometimes that includes dead ends ...
- ... but users can be taught about that, too.

Avoid high entry barriers

- GAP is a really cool project, with a friendly and helpful community!
- But to an outsider, GAP *development* is opaque ☹
- GAP download page:
- Contrary to appearance, GAP is very much alive ...
- ... but if you are not already “inside”, there is no way of knowing that ☹
- The only public reference to GAP CVS I could find is
`http://www.gap-system.org/oldsite/Info4/cvs.html`
- So how do you get (read) access to the GAP CVS?
- Just ask! But... do you dare to? Who is eligible?
- ~> high entry barrier for potential contributors

My dream: Open repository, bug tracker, development discussion, ...

Avoid high entry barriers

- GAP is a really cool project, with a friendly and helpful community!
- But to an outsider, GAP *development* is opaque ☹
- GAP download page:
- Contrary to appearance, GAP is very much alive ...
- ... but if you are not already “inside”, there is no way of knowing that ☹
- The only public reference to GAP CVS I could find is
`http://www.gap-system.org/oldsite/Info4/cvs.html`
- So how do you get (read) access to the GAP CVS?
- Just ask! But... do you dare to? Who is eligible?
- ~> high entry barrier for potential contributors

My dream: Open repository, bug tracker, development discussion, ...

Avoid high entry barriers

- GAP is a really cool project, with a friendly and helpful community!
- But to an outsider, GAP *development* is opaque ☹
- GAP download page:
 - Contrary to appearance, GAP is very much alive ...
 - ... but if you are not already “inside”, there is no way of knowing that ☹
 - The only public reference to GAP CVS I could find is <http://www.gap-system.org/oldsite/Info4/cvs.html>
 - So how do you get (read) access to the GAP CVS?
 - Just ask! But... do you dare to? Who is eligible?
 - ~> high entry barrier for potential contributors

My dream: Open repository, bug tracker, development discussion, ...

Avoid high entry barriers

- GAP is a really cool project, with a friendly and helpful community!
- But to an outsider, GAP *development* is opaque ☹️
- GAP download page: **The currently released version of GAP is 4.4.12 of December 2008.**
- Contrary to appearance, GAP is very much alive ...
- ... but if you are not already “inside”, there is no way of knowing that ☹️
- The only public reference to GAP CVS I could find is
`http://www.gap-system.org/oldsite/Info4/cvs.html`
- So how do you get (read) access to the GAP CVS?
- Just ask! But... do you dare to? Who is eligible?
- ~> high entry barrier for potential contributors

My dream: Open repository, bug tracker, development discussion, ...

Avoid high entry barriers

- GAP is a really cool project, with a friendly and helpful community!
- But to an outsider, GAP *development* is opaque ☹️
- GAP download page: **The currently released version of GAP is 4.4.12 of December 2008.**
- Contrary to appearance, GAP is very much alive ...
- ... but if you are not already “inside”, there is no way of knowing that ☹️
- The only public reference to GAP CVS I could find is
`http://www.gap-system.org/oldsite/Info4/cvs.html`
- So how do you get (read) access to the GAP CVS?
- Just ask! But... do you dare to? Who is eligible?
- ~> high entry barrier for potential contributors

My dream: Open repository, bug tracker, development discussion, ...

Avoid high entry barriers

- GAP is a really cool project, with a friendly and helpful community!
- But to an outsider, GAP *development* is opaque ☹
- GAP download page: **The currently released version of GAP is 4.4.12 of December 2008.**
- Contrary to appearance, GAP is very much alive ...
- ... but if you are not already “inside”, there is no way of knowing that ☹
- The only public reference to GAP CVS I could find is
`http://www.gap-system.org/oldsite/Info4/cvs.html`
- So how do you get (read) access to the GAP CVS?
- Just ask! But... do you dare to? Who is eligible?
- ~> high entry barrier for potential contributors

My dream: Open repository, bug tracker, development discussion, ...

Avoid high entry barriers

- GAP is a really cool project, with a friendly and helpful community!
- But to an outsider, GAP *development* is opaque ☹
- GAP download page: **The currently released version of GAP is 4.4.12 of December 2008.**
- Contrary to appearance, GAP is very much alive ...
- ... but if you are not already “inside”, there is no way of knowing that ☹
- The only public reference to GAP CVS I could find is
<http://www.gap-system.org/oldsite/Info4/cvs.html>
- So how do you get (read) access to the GAP CVS?
- Just ask! But... do you dare to? Who is eligible?
- ~> high entry barrier for potential contributors

My dream: Open repository, bug tracker, development discussion, ...

Avoid high entry barriers

- GAP is a really cool project, with a friendly and helpful community!
- But to an outsider, GAP *development* is opaque ☹
- GAP download page: **The currently released version of GAP is 4.4.12 of December 2008.**
- Contrary to appearance, GAP is very much alive ...
- ... but if you are not already “inside”, there is no way of knowing that ☹
- The only public reference to GAP CVS I could find is
<http://www.gap-system.org/oldsite/Info4/cvs.html>
- So how do you get (read) access to the GAP CVS?
 - Just ask! But... do you dare to? Who is eligible?
 - ~> high entry barrier for potential contributors

My dream: Open repository, bug tracker, development discussion, ...

Avoid high entry barriers

- GAP is a really cool project, with a friendly and helpful community!
- But to an outsider, GAP *development* is opaque ☹
- GAP download page: **The currently released version of GAP is 4.4.12 of December 2008.**
- Contrary to appearance, GAP is very much alive ...
- ... but if you are not already “inside”, there is no way of knowing that ☹
- The only public reference to GAP CVS I could find is
<http://www.gap-system.org/oldsite/Info4/cvs.html>
- So how do you get (read) access to the GAP CVS?
- Just ask! But... do you dare to? Who is eligible?
- ~> high entry barrier for potential contributors

My dream: Open repository, bug tracker, development discussion, ...

Avoid high entry barriers

- GAP is a really cool project, with a friendly and helpful community!
- But to an outsider, GAP *development* is opaque ☹
- GAP download page: **The currently released version of GAP is 4.4.12 of December 2008.**
- Contrary to appearance, GAP is very much alive ...
- ... but if you are not already “inside”, there is no way of knowing that ☹
- The only public reference to GAP CVS I could find is
`http://www.gap-system.org/oldsite/Info4/cvs.html`
- So how do you get (read) access to the GAP CVS?
- Just ask! But... do you dare to? Who is eligible?
- ~> high entry barrier for potential contributors

My dream: Open repository, bug tracker, development discussion, ...

Avoid high entry barriers

- GAP is a really cool project, with a friendly and helpful community!
- But to an outsider, GAP *development* is opaque ☹️
- GAP download page: **The currently released version of GAP is 4.4.12 of December 2008.**
- Contrary to appearance, GAP is very much alive ...
- ... but if you are not already “inside”, there is no way of knowing that ☹️
- The only public reference to GAP CVS I could find is
`http://www.gap-system.org/oldsite/Info4/cvs.html`
- So how do you get (read) access to the GAP CVS?
- Just ask! But... do you dare to? Who is eligible?
- ~> high entry barrier for potential contributors

My dream: Open repository, bug tracker, development discussion, ...

Overview

- My background
- What does “open source” mean, anyway?
- Why should you care?
- How to be open and make the best out of it?
- **Tools, tools, tools**

Some useful tools for the development process

- **Version control:** CVS, Subversion, Git, Mercurial, Bazaar, ...
- **Bug tracking:** Bugzilla, Trac, Redmine, ...; Lighthouse, Jira, ...
- **Wiki:** MediaWiki, Trac, Redmine, ...
- **Forums & mailing lists:** Mailman, Redmine, phpBB, ...; Google Groups, Yahoo Groups, Tender, ...
- **Continuous integration:** BuildBot, CruiseControl, Jenkins (Hudson), ...

Test driven development; Extreme programming; Agile development; other software engineering ideas \leadsto another talk ☺

Tools that help with the development itself, e.g. build tools (autoconf, automake, CMake, scons, ...); useful libraries (e.g. GMP) \leadsto more talks ☺

Some useful tools for the development process

- **Version control:** CVS, Subversion, Git, Mercurial, Bazaar, ...
- **Bug tracking:** Bugzilla, Trac, Redmine, ...; Lighthouse, Jira, ...
- **Wiki:** MediaWiki, Trac, Redmine, ...
- **Forums & mailing lists:** Mailman, Redmine, phpBB, ...; Google Groups, Yahoo Groups, Tender, ...
- **Continuous integration:** BuildBot, CruiseControl, Jenkins (Hudson), ...

Test driven development; Extreme programming; Agile development; other software engineering ideas \leadsto another talk ☺

Tools that help with the development itself, e.g. build tools (autoconf, automake, CMake, scons, ...); useful libraries (e.g. GMP) \leadsto more talks ☺

Some useful tools for the development process

- **Version control:** CVS, Subversion, Git, Mercurial, Bazaar, ...
- **Bug tracking:** Bugzilla, Trac, Redmine, ...; Lighthouse, Jira, ...
- **Wiki:** MediaWiki, Trac, Redmine, ...
- **Forums & mailing lists:** Mailman, Redmine, phpBB, ...; Google Groups, Yahoo Groups, Tender, ...
- **Continuous integration:** BuildBot, CruiseControl, Jenkins (Hudson), ...

Test driven development; Extreme programming; Agile development; other software engineering ideas \leadsto another talk ☺

Tools that help with the development itself, e.g. build tools (autoconf, automake, CMake, scons, ...); useful libraries (e.g. GMP) \leadsto more talks ☺

Some useful tools for the development process

- **Version control:** CVS, Subversion, Git, Mercurial, Bazaar, ...
- **Bug tracking:** Bugzilla, Trac, Redmine, ...; Lighthouse, Jira, ...
- **Wiki:** MediaWiki, Trac, Redmine, ...
- **Forums & mailing lists:** Mailman, Redmine, phpBB, ...; Google Groups, Yahoo Groups, Tender, ...
- **Continuous integration:** BuildBot, CruiseControl, Jenkins (Hudson), ...

Test driven development; Extreme programming; Agile development; other software engineering ideas \leadsto another talk ☺

Tools that help with the development itself, e.g. build tools (autoconf, automake, CMake, scons, ...); useful libraries (e.g. GMP) \leadsto more talks ☺

Some useful tools for the development process

- **Version control:** CVS, Subversion, Git, Mercurial, Bazaar, ...
- **Bug tracking:** Bugzilla, Trac, Redmine, ...; Lighthouse, Jira, ...
- **Wiki:** MediaWiki, Trac, Redmine, ...
- **Forums & mailing lists:** Mailman, Redmine, phpBB, ...; Google Groups, Yahoo Groups, Tender, ...
- **Continuous integration:** BuildBot, CruiseControl, Jenkins (Hudson), ...

Test driven development; Extreme programming; Agile development; other software engineering ideas \leadsto another talk ☺

Tools that help with the development itself, e.g. build tools (autoconf, automake, CMake, scons, ...); useful libraries (e.g. GMP) \leadsto more talks ☺

Some useful tools for the development process

- **Version control:** CVS, Subversion, Git, Mercurial, Bazaar, ...
- **Bug tracking:** Bugzilla, Trac, Redmine, ...; Lighthouse, Jira, ...
- **Wiki:** MediaWiki, Trac, Redmine, ...
- **Forums & mailing lists:** Mailman, Redmine, phpBB, ...; Google Groups, Yahoo Groups, Tender, ...
- **Continuous integration:** BuildBot, CruiseControl, Jenkins (Hudson), ...

Test driven development; Extreme programming; Agile development; other software engineering ideas \leadsto another talk ☺

Tools that help with the development itself, e.g. build tools (autoconf, automake, CMake, scons, ...); useful libraries (e.g. GMP) \leadsto more talks ☺

Some useful tools for the development process

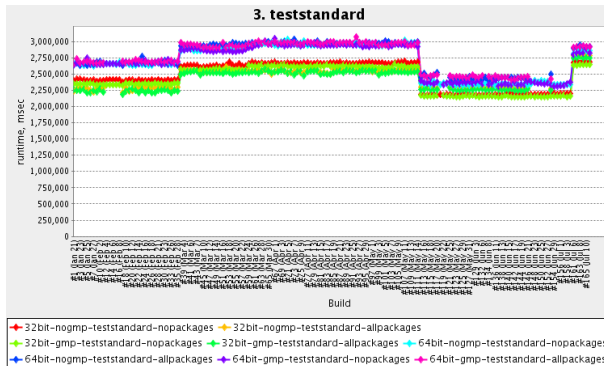
- **Version control:** CVS, Subversion, Git, Mercurial, Bazaar, ...
- **Bug tracking:** Bugzilla, Trac, Redmine, ...; Lighthouse, Jira, ...
- **Wiki:** MediaWiki, Trac, Redmine, ...
- **Forums & mailing lists:** Mailman, Redmine, phpBB, ...; Google Groups, Yahoo Groups, Tender, ...
- **Continuous integration:** BuildBot, CruiseControl, Jenkins (Hudson), ...

Test driven development; Extreme programming; Agile development; other software engineering ideas \rightsquigarrow another talk ☺

Tools that help with the development itself, e.g. build tools (autoconf, automake, CMake, scon, ...); useful libraries (e.g. GMP) \rightsquigarrow more talks ☺

Example: Version control + continuous integration = win

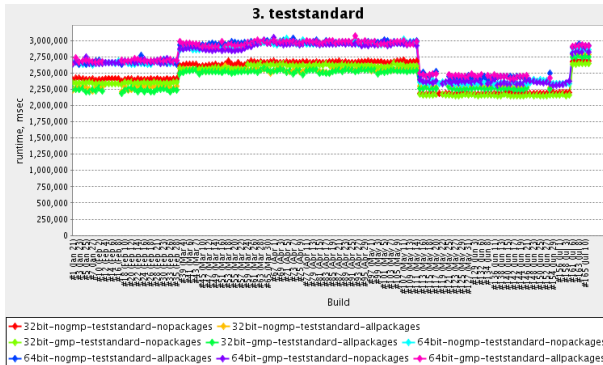
- The GAP team uses a Jenkins server for continuous integration.
- This way, regressions can be quickly noticed and resolved.



- A build server can also be used to provide daily builds for alpha testers.

Example: Version control + continuous integration = win

- The GAP team uses a Jenkins server for continuous integration.
- This way, regressions can be quickly noticed and resolved.



- A build server can also be used to provide daily builds for alpha testers.

A plea for using hosted collaboration tools

- If you are author of a small package (e.g. for GAP), please consider using a hosting provider (like github, SourceForge, etc.) for your project!
- If you host the project in your personal webspace, then ...
- ... whenever you move to a new job, your project has to move;
- ... if you loose time / interest, nobody else can update it.
- If you collect bug reports in your mailbox alone, they are lost if you leave the project \leadsto use a public bug tracker.
- Nowadays, you don't have to be big (like GAP, Singular, etc.) with your own servers in order to have a a VCS, a public bug tracker, mailing lists, forums, a wiki, and more!

A plea for using hosted collaboration tools

- If you are author of a small package (e.g. for GAP), please consider using a hosting provider (like github, SourceForge, etc.) for your project!
- If you host the project in your personal webspace, then ...
 - ... whenever you move to a new job, your project has to move;
 - ... if you loose time / interest, nobody else can update it.
- If you collect bug reports in your mailbox alone, they are lost if you leave the project ~> use a public bug tracker.
- Nowadays, you don't have to be big (like GAP, Singular, etc.) with your own servers in order to have a a VCS, a public bug tracker, mailing lists, forums, a wiki, and more!

A plea for using hosted collaboration tools

- If you are author of a small package (e.g. for GAP), please consider using a hosting provider (like github, SourceForge, etc.) for your project!
- If you host the project in your personal webspace, then ...
- ... whenever you move to a new job, your project has to move;
- ... if you loose time / interest, nobody else can update it.
- If you collect bug reports in your mailbox alone, they are lost if you leave the project ~> use a public bug tracker.
- Nowadays, you don't have to be big (like GAP, Singular, etc.) with your own servers in order to have a a VCS, a public bug tracker, mailing lists, forums, a wiki, and more!

A plea for using hosted collaboration tools

- If you are author of a small package (e.g. for GAP), please consider using a hosting provider (like github, SourceForge, etc.) for your project!
- If you host the project in your personal webspace, then ...
- ... whenever you move to a new job, your project has to move;
- ... if you loose time / interest, nobody else can update it.
- If you collect bug reports in your mailbox alone, they are lost if you leave the project \leadsto use a public bug tracker.
- Nowadays, you don't have to be big (like GAP, Singular, etc.) with your own servers in order to have a a VCS, a public bug tracker, mailing lists, forums, a wiki, and more!

A plea for using hosted collaboration tools

- If you are author of a small package (e.g. for GAP), please consider using a hosting provider (like github, SourceForge, etc.) for your project!
- If you host the project in your personal webspace, then ...
- ... whenever you move to a new job, your project has to move;
- ... if you loose time / interest, nobody else can update it.
- If you collect bug reports in your mailbox alone, they are lost if you leave the project \leadsto use a public bug tracker.
- Nowadays, you don't have to be big (like GAP, Singular, etc.) with your own servers in order to have a a VCS, a public bug tracker, mailing lists, forums, a wiki, and more!

A plea for using hosted collaboration tools

- If you are author of a small package (e.g. for GAP), please consider using a hosting provider (like github, SourceForge, etc.) for your project!
- If you host the project in your personal webspace, then ...
- ... whenever you move to a new job, your project has to move;
- ... if you loose time / interest, nobody else can update it.
- If you collect bug reports in your mailbox alone, they are lost if you leave the project \leadsto use a public bug tracker.
- Nowadays, you don't have to be big (like GAP, Singular, etc.) with your own servers in order to have a a VCS, a public bug tracker, mailing lists, forums, a wiki, and more!

Some open source hosting platforms

Some sites offer free “all-included” setups for open source projects (including version control, issue tracking, Wikis and more):

- [GitHub](http://github.com) (<http://github.com>)
- [SourceForge](http://sf.net) (<http://sf.net>)
- [Bitbucket](http://bitbucket.org) (<http://bitbucket.org>)
- [Google Code](http://code.google.com/hosting) (<http://code.google.com/hosting>)
- [Assembla](http://www.assembla.com) (<http://www.assembla.com>)
- [Launchpad](https://launchpad.net) (<https://launchpad.net>)
- ...

Some open source hosting platforms

Some sites offer free “all-included” setups for open source projects (including version control, issue tracking, Wikis and more):

- **GitHub** (<http://github.com>)
- **SourceForge** (<http://sf.net>)
- **Bitbucket** (<http://bitbucket.org>)
- **Google Code** (<http://code.google.com/hosting>)
- **Assembla** (<http://www.assembla.com>)
- **Launchpad** (<https://launchpad.net>)
- ...

Some open source hosting platforms

Some sites offer free “all-included” setups for open source projects (including version control, issue tracking, Wikis and more):

- **GitHub** (<http://github.com>)
- **SourceForge** (<http://sf.net>)
- **Bitbucket** (<http://bitbucket.org>)
- **Google Code** (<http://code.google.com/hosting>)
- **Assembla** (<http://www.assembla.com>)
- **Launchpad** (<https://launchpad.net>)
- ...

Some open source hosting platforms

Some sites offer free “all-included” setups for open source projects (including version control, issue tracking, Wikis and more):

- **GitHub** (<http://github.com>)
- **SourceForge** (<http://sf.net>)
- **Bitbucket** (<http://bitbucket.org>)
- **Google Code** (<http://code.google.com/hosting>)
- **Assembla** (<http://www.assembla.com>)
- **Launchpad** (<https://launchpad.net>)
- ...

Some open source hosting platforms

Some sites offer free “all-included” setups for open source projects (including version control, issue tracking, Wikis and more):

- **GitHub** (<http://github.com>)
- **SourceForge** (<http://sf.net>)
- **Bitbucket** (<http://bitbucket.org>)
- **Google Code** (<http://code.google.com/hosting>)
- **Assembla** (<http://www.assembla.com>)
- **Launchpad** (<https://launchpad.net>)
- ...

Some open source hosting platforms

Some sites offer free “all-included” setups for open source projects (including version control, issue tracking, Wikis and more):

- **GitHub** (<http://github.com>)
- **SourceForge** (<http://sf.net>)
- **Bitbucket** (<http://bitbucket.org>)
- **Google Code** (<http://code.google.com/hosting>)
- **Assembla** (<http://www.assembla.com>)
- **Launchpad** (<https://launchpad.net>)
- ...

Some open source hosting platforms

Some sites offer free “all-included” setups for open source projects (including version control, issue tracking, Wikis and more):

- **GitHub** (<http://github.com>)
- **SourceForge** (<http://sf.net>)
- **Bitbucket** (<http://bitbucket.org>)
- **Google Code** (<http://code.google.com/hosting>)
- **Assembla** (<http://www.assembla.com>)
- **Launchpad** (<https://launchpad.net>)
- . . .

Some open source hosting platforms

Some sites offer free “all-included” setups for open source projects (including version control, issue tracking, Wikis and more):

- **GitHub** (<http://github.com>)
- **SourceForge** (<http://sf.net>)
- **Bitbucket** (<http://bitbucket.org>)
- **Google Code** (<http://code.google.com/hosting>)
- **Assembla** (<http://www.assembla.com>)
- **Launchpad** (<https://launchpad.net>)
- ...

Live demo

Your tools are useless if nobody knows about them

- Once you have a version control system, a Wiki, a build server, etc., make sure that people find them!
- It is an annoying loss if new (potential and actual) developers do not know about all the available resources.
- Example: ScummVM collects this information on a set of web pages:
http://wiki.scummvm.org/index.php/Project_Services

Your tools are useless if nobody knows about them

- Once you have a version control system, a Wiki, a build server, etc., make sure that people find them!
- It is an annoying loss if new (potential and actual) developers do not know about all the available resources.
- Example: ScummVM collects this information on a set of web pages:
http://wiki.scummvm.org/index.php/Project_Services

Your tools are useless if nobody knows about them

- Once you have a version control system, a Wiki, a build server, etc., make sure that people find them!
- It is an annoying loss if new (potential and actual) developers do not know about all the available resources.
- Example: ScummVM collects this information on a set of web pages:
http://wiki.scummvm.org/index.php/Project_Services


SVM project services

Excerpt from

http://wiki.scummvm.org/index.php/Project_Services


BuildBot

Main article: [Buildbot](#)

- <http://buildbot.scummvm.org/> 
- Purpose: Provides automated build services for an increasing number of our supported platforms.
- Primary Maintainer(s): Andre Heider ([Dhewg](#)) | John Willis ([DJWillis](#))


Git

Main article: [Git](#)

- <https://github.com/scummvm/scummvm/> 
- Purpose: Store the project's source code, keeping the development history and allowing collaborative work.
- Primary Maintainer(s): Eugene Sandulenko ([Sev](#))

Doxygen Project Documentation

Main article: [Doxygen](#)

- <http://doxygen.scummvm.org/> 
- Purpose: This aims to provide current, automatically generated, documentation from our source code.
- Primary Maintainer(s): Joost Peters ([JoostP](#))

Mailing list

Main article: [Mailing lists](#)

- <https://lists.sourceforge.net/lists/admin/db/scummvm-devel> 
- Purpose: Official developer mailing lists, for planning releases, and general discussions concerning development and project management.
- Primary Maintainer(s): Eugene Sandulenko ([Sev](#))

⋮

That's it for today!

The End?

References

- ScummVM code size graph take from Ohloh
- Arduino picture taken from Wikipedia
- Free Beer picture taken from Wikipedia
- The Cathedral and the Bazaar by Eric S. Raymond (see also Wikipedia)