redhat.

How to start an open source project of any size and scope

Karsten Wade, Sr. Community Architect

Red Hat Community Architecture & Leadership Team

This presentation: http://bit.ly/SCALE10x_HowToStartOSS

Agenda

- What is TheOpenSourceWay.org & how is it related?
- How does project size & origin come in to play?
- All the steps in tl;dnr:
  - Steps to start a project.
  - Steps to sustain a project.
  - Steps to grow a project.
- Conclusions.
- Questions and discussion.
- Anyone here we can help?

What is TheOpenSourceWay.org?

- Community book written the community way.
- Upstream for useful content.
- Handbook, methods, checklist.
  - Incomplete encourages participation.
- The open source way distilled.
  - Cf. Producing OSS (Fogel).

Considerations for project size and origin

- Generally, all principles apply the same.
  - But size and origin do have special needs.

Considerations for project size and origin

- Generally, all principles apply the same.
  - But size and origin do have special needs.
- Large projects are different because …
  - Budget approval.
  - Vendors like the Big Reveal.
  - Need to be modular.
  - Diff. from single person w/ large userbase.

Considerations for project size and origin

- Generally, all principles apply the same.
    - But size and origin do have special needs.
- Large projects are different because …
    - Budget approval.
    - Vendors like the Big Reveal.
    - Need to be modular.
    - Diff. from single person w/ large userbase.
- Small projects are different because …
    - Nimble but hard to do it all yourself.
    - People can see themselves in the project more easily.

Considerations for project size and origin (cont'd)

- Sponsored-by-large-org projects are different because …
  - Stigma & worries about abuse of power.
  - Governance is highly important.
  - Be careful with marketing machine.

Considerations for project size and origin (cont'd)

- Sponsored-by-large-org projects are different because …
  - Stigma & worries about abuse of power.
  - Governance is highly important.
  - Be careful with marketing machine.
- Academic-sourced projects are different because ...
  - Small with a potential Big Machine behind it.
  - Graduation risk – continuity needed.

Considerations for project size and origin (cont'd)

- Sponsored-by-large-org projects are different because …
  - Stigma & worries about abuse of power.
  - Governance is highly important.
  - Be careful with marketing machine.
- Academic-sourced projects are different because ...
  - Small with a potential Big Machine behind it.
  - Graduation risk – continuity needed.
- Government projects are different because …
  - Must require contractors to open source code.
  - Procurement rules require COTS.
  - Privacy, security, other issues require diff thinking.

Starting steps in tl;dnr format

1. Initial governance.

2. Contribution policy.

3. External main project mailing list.

4. Source control for code and content.

5. Issue tracker is a general tool or method for the community to keep track of important issues (projects, problems, tasks) in a central way.

6. Wiki for community, collaborative documentation.

7. Weekly IRC meeting time.

8. Team planet/blog feed.

9. Open roadmap for the project on the wiki.

10. Simple open marketing plan, posted on project wiki, talked about on main mailing list – events, people to focus on, press, online seminars, etc.

11. Expose interesting and easier tasks – encourage peripheral participation.

12. Volunteer mentors wiki page.

13. How to participate and contribute page.

14. Community information page – including leadership and press contacts.

15. Participant and contributor improvements and needs page - wish list and roadmap for how things can/should improve for contributors and participants, over time.

Starting steps

1. Initial governance.

Starting steps

1. Initial governance.
2. Contribution policy.

Starting steps

1. Initial governance.
2. Contribution policy.
3. External main project mailing list.

Starting steps

1. Initial governance.

2. Contribution policy.

3. External main project mailing list.

4. Source control for code and content.

Starting steps

1. Initial governance.
2. Contribution policy.
3. External main project mailing list.
4. Source control for code and content.
5. Issue tracker is a general tool or method for the community to keep track of important issues (projects, problems, tasks) in a central way.

Starting steps

1. Initial governance.
2. Contribution policy.
3. External main project mailing list.
4. Source control for code and content.
5. Issue tracker is a general tool or method for the community to keep track of important issues (projects, problems, tasks) in a central way.
6. Wiki for community, collaborative documentation.

Starting steps

1. Initial governance.
2. Contribution policy.
3. External main project mailing list.
4. Source control for code and content.
5. Issue tracker is a general tool or method for the community to keep track of important issues (projects, problems, tasks) in a central way.
6. Wiki for community, collaborative documentation.
7. Weekly IRC meeting time.

Starting steps

1. Initial governance.
2. Contribution policy.
3. External main project mailing list.
4. Source control for code and content.
5. Issue tracker is a general tool or method for the community to keep track of important issues (projects, problems, tasks) in a central way.
6. Wiki for community, collaborative documentation.
7. Weekly IRC meeting time.
8. Team planet/blog feed.

Starting steps

1. Initial governance.
2. Contribution policy.
3. External main project mailing list.
4. Source control for code and content.
5. Issue tracker is a general tool or method for the community to keep track of important issues (projects, problems, tasks) in a central way.
6. Wiki for community, collaborative documentation.
7. Weekly IRC meeting time.
8. Team planet/blog feed.
9. Open roadmap for the project on the wiki.

Starting steps (cont'd)

10. Simple open marketing plan, posted on project wiki, talked about on main mailing list – events, people to focus on, press, online seminars, etc.

Starting steps (cont'd)

10. Simple open marketing plan, posted on project wiki, talked about on main mailing list – events, people to focus on, press, online seminars, etc.

11. Expose interesting and easier tasks – encourage peripheral participation.

Starting steps (cont'd)

10. Simple open marketing plan, posted on project wiki, talked about on main mailing list – events, people to focus on, press, online seminars, etc.
11. Expose interesting and easier tasks – encourage peripheral participation.
12. Volunteer mentors wiki page.

Starting steps (cont'd)

10. Simple open marketing plan, posted on project wiki, talked about on main mailing list – events, people to focus on, press, online seminars, etc.

11. Expose interesting and easier tasks – encourage peripheral participation.

12. Volunteer mentors wiki page.

13. How to participate and contribute page.

Starting steps (cont'd)

10. Simple open marketing plan, posted on project wiki, talked about on main mailing list – events, people to focus on, press, online seminars, etc.
11. Expose interesting and easier tasks – encourage peripheral participation.
12. Volunteer mentors wiki page.
13. How to participate and contribute page.
14. Community information page – including leadership and press contacts.

Starting steps (cont'd)

10. Simple open marketing plan, posted on project wiki, talked about on main mailing list – events, people to focus on, press, online seminars, etc.
11. Expose interesting and easier tasks – encourage peripheral participation.
12. Volunteer mentors wiki page.
13. How to participate and contribute page.
14. Community information page – including leadership and press contacts.
15. Participant and contributor improvements and needs page - wish list and roadmap for how things can/should improve for contributors and participants, over time.

Sustaining steps in tl;dnr format

1. Get out of the way & let it thrive.

2. Improve your infrastructure - focus on enabling contributions.

3. Remember what you learned in Kindergarten - be nice, share.

4. Focus on legitimate peripheral participation.

5. All discussions and decisions must default to OPEN.

6. Use version control for content & code.

7. Choose open tools that can be extended.

8. Focus on keeping the community healthy.

9. Keep governance real and active.

10. Deal with poison.

11. Ongoing communicator(s) are needed.

12. Consensus over voting.

13. Release early, release often.

14. Use a predictable schedule, stick to it.

Sustaining steps

1. Get out of the way & let it thrive.

Sustaining steps

1. Get out of the way & let it thrive.
2. Improve your infrastructure - focus on enabling contributions.

Sustaining steps

1. Get out of the way & let it thrive.
2. Improve your infrastructure - focus on enabling contributions.
3. Remember what you learned in Kindergarten - be nice, share.

Sustaining steps

1. Get out of the way & let it thrive.
2. Improve your infrastructure - focus on enabling contributions.
3. Remember what you learned in Kindergarten - be nice, share.
4. Focus on legitimate peripheral participation.

Sustaining steps

1. Get out of the way & let it thrive.
2. Improve your infrastructure - focus on enabling contributions.
3. Remember what you learned in Kindergarten - be nice, share.
4. Focus on legitimate peripheral participation.
5. All discussions and decisions must default to OPEN.

Sustaining steps

1. Get out of the way & let it thrive.
2. Improve your infrastructure - focus on enabling contributions.
3. Remember what you learned in Kindergarten - be nice, share.
4. Focus on legitimate peripheral participation.
5. All discussions and decisions must default to OPEN.
6. Use version control for content & code.

Sustaining steps

1. Get out of the way & let it thrive.
2. Improve your infrastructure - focus on enabling contributions.
3. Remember what you learned in Kindergarten - be nice, share.
4. Focus on legitimate peripheral participation.
5. All discussions and decisions must default to OPEN.
6. Use version control for content & code.
7. Choose open tools that can be extended.

Sustaining steps (cont'd)

8. Focus on keeping the community healthy.

Sustaining steps (cont'd)

8. Focus on keeping the community healthy.

9. Keep governance real and active.

Sustaining steps (cont'd)

8. Focus on keeping the community healthy.

9. Keep governance real and active.

10. Deal with poison.

Sustaining steps (cont'd)

8. Focus on keeping the community healthy.

9. Keep governance real and active.

10. Deal with poison.

11. Ongoing communicator(s) are needed.

Sustaining steps (cont'd)

8. Focus on keeping the community healthy.

9. Keep governance real and active.

10. Deal with poison.

11. Ongoing communicator(s) are needed.

12. Consensus over voting.

Sustaining steps (cont'd)

8. Focus on keeping the community healthy.

9. Keep governance real and active.

10. Deal with poison.

11. Ongoing communicator(s) are needed.

12. Consensus over voting.

13. Release early, release often.

Sustaining steps (cont'd)

8. Focus on keeping the community healthy.

9. Keep governance real and active.

10. Deal with poison.

11. Ongoing communicator(s) are needed.

12. Consensus over voting.

13. Release early, release often.

14. Use a predictable schedule, stick to it.

Growth steps

1. Evolve with community.

Growth steps

1. Evolve with community.
2. Change leadership regularly, democratically-by-merit.

Growth steps

1. Evolve with community.
2. Change leadership regularly, democratically-by-merit.
3. Be students as well as teacher/mentors.

Growth steps

1. Evolve with community.
2. Change leadership regularly, democratically-by-merit.
3. Be students as well as teacher/mentors.
4. Drive down barriers.

Growth steps

1. Evolve with community.
2. Change leadership regularly, democratically-by-merit.
3. Be students as well as teacher/mentors.
4. Drive down barriers.
5. Embrace failure.

Growth steps

1. Evolve with community.

2. Change leadership regularly, democratically-by-merit.

3. Be students as well as teacher/mentors.

4. Drive down barriers.

5. Embrace failure.

6. ... and more!

- This is an example of where TheOpenSourceWay.org is incomplete and needs more, more, more.

Let's talk about the risks and rewards of growth ...

- Risks:
  - Growing too fast, too big, too diverse.
  - Loss of focus.
  - Dilution of vision.
  - Flamewars & bikeshedding.

Let's talk about the risks and rewards of growth ...

- Risks:
  - Growing too fast, too big, too diverse.
  - Loss of focus.
  - Dilution of vision.
  - Flamewars & bikeshedding.
- Rewards:
  - Take work off peoples' shoulders.
  - Larger, more creative innovation stream.
  - Greater reach locally and globally.

![Red Hat logo](red hat)

Let's talk about the risks and rewards of growth ...

- Risks:
  - Growing too fast, too big, too diverse.
  - Loss of focus.
  - Dilution of vision.
  - Flamewars & bikeshedding.
- Rewards:
  - Take work off peoples' shoulders.
  - Larger, more creative innovation stream.
  - Greater reach locally and globally.
- What is the "right size" for your project?

Conclusions?

Questions?

Discussion?

Anyone here need some help today?

How to start an open source project of
any size and scope

Karsten Wade, Sr. Community Architect

Red Hat Community Architecture & Leadership Team

This presentation: http://bit.ly/SCALE10x_HowToStartOSS

Why am I giving this presentation? I see a lot of articles and such on this topic, I
tend to find them not very comprehensive, and I'd really like to show the value of
the work done on theopensourceway.org in this regard.

Who am I?

redhat

Agenda

- What is TheOpenSourceWay.org & how is it related?
- How does project size & origin come in to play?
- All the steps in tl;dnr:
  - Steps to start a project.
  - Steps to sustain a project.
  - Steps to grow a project.
- Conclusions.
- Questions and discussion.
- Anyone here we can help?

On the last item, I'm wondering if some of you here are thinking of starting, already have started, or want to look at how to improve your own open source projects. I'd like to see if we can help you here more than just answer questions. For example, if you discover in this process that you forgot to put a contribution policy on your wiki, we can work here on the language and get it posted before the hour is up. So think on that as we talk.

redhat.

What is TheOpenSourceWay.org?

- Community book written the community way.
- Upstream for useful content.
- Handbook, methods, checklist.
  - Incomplete encourages participation.
- The open source way distilled.
  - Cf. Producing OSS (Fogel).

Considerations for project size and origin

- Generally, all principles apply the same.
  - But size and origin do have special needs.

Starting a project as large from the beginning is a huge challenge. You need budget, meaning approval from someone, or someone is you, justification to the board/shareholders/partners. Often people are drawn to making a large project for the same reason they like the Big Reveal. It's part of the way vendors do things. It helps if you can find ways to modularize the project, so each module (sub-project) can thrive and grow on its own. Cf. Apache, Linux Kernel However, it's possible for one person to start something that is effectively huge, such as their own Linux distro. This is really just a small project+ (one or a few people) with a large goal and audience potential.

Small projects have all the usual advantages of being nimble and able to recover from adversity. The disads include the usual hard to do it much with few people, get attention, etc. In FOSS, however, small has special powers. You can attract people who have skills your project needs because people can see themselves part of something smaller more easily than part of a Big Machine. Projects such as Fedora work extra hard to make sure people know they can participate, how to do it, etc. In those cases, you may help 50 people to get one contributor. For a small project, no science here but just instinct says, it's easier to reach people and get them involved in something that they can hold in their hand.

4

redhat.

Considerations for project size and origin

- Generally, all principles apply the same.
  - But size and origin do have special needs.
- Large projects are different because …
- Budget approval.
- Vendors like the Big Reveal.
- Need to be modular.
- Diff. from single person w/ large userbase.

Starting a project as large from the beginning is a huge challenge.You need budget, meaning approval from someone, or someone is you, justification to the board/shareholders/partners. Often people are drawn to making a large project for the same reason they like the Big Reveal. It's part of the way vendors do things. It helps if you can find ways to modularize the project, so each module (sub-project) can thrive and grow on its own. Cf. Apache, Linux Kernel However, it's possible for one person to start something that is effectively huge, such as their own Linux distro. This is really just a small project+ (one or a few people) with a large goal and audience potential.

Small projects have all the usual advantages of being nimble and able to recover from adversity. The disads include the usual hard to do it much with few people, get attention, etc. In FOSS, however, small has special powers. You can attract people who have skills your project needs because people can see themselves part of something smaller more easily than part of a Big Machine. Projects such as Fedora work extra hard to make sure people know they can participate, how to do it, etc. In those cases, you may help 50 people to get one contributor. For a small project, no science here but just instinct says, it's easier to reach people and get them involved in something that they can hold in their hand.
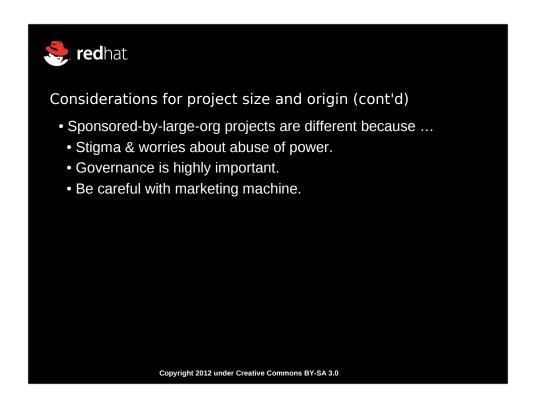
Considerations for project size and origin

- Generally, all principles apply the same.
  - But size and origin do have special needs.
- Large projects are different because …
  - Budget approval.
  - Vendors like the Big Reveal.
  - Need to be modular.
  - Diff. from single person w/ large userbase.
- Small projects are different because …
  - Nimble but hard to do it all yourself.
  - People can see themselves in the project
    more easily.

Starting a project as large from the beginning is a huge challenge.You need budget, meaning approval from someone, or someone is you, justification to the board/shareholders/partners. Often people are drawn to making a large project for the same reason they like the Big Reveal. It's part of the way vendors do things. It helps if you can find ways to modularize the project, so each module (sub-project) can thrive and grow on its own. Cf. Apache, Linux Kernel However, it's possible for one person to start something that is effectively huge, such as their own Linux distro. This is really just a small project+ (one or a few people) with a large goal and audience potential.
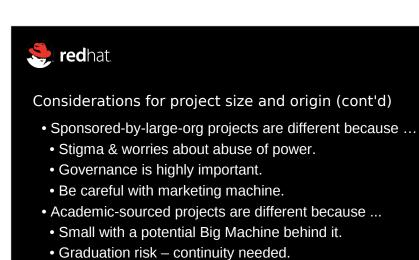
Small projects have all the usual advantages of being nimble and able to recover from adversity. The disads include the usual hard to do it much with few people, get attention, etc. In FOSS, however, small has special powers. You can attract people who have skills your project needs because people can see themselves part of something smaller more easily than part of a Big Machine. Projects such as Fedora work extra hard to make sure people know they can participate, how to do it, etc. In those cases, you may help 50 people to get one contributor. For a small project, no science here but just instinct says, it's easier to reach people and get them involved in something that they can hold in their hand.

Stigma of the large org. People worry about abuses of power. Quality governance is super-important - people have to see how it's enshrined that the large-org isn't going to abuse power. It's easy to have the marketing machine get involved, which can help boost attention. Be very careful that the initial steps are done - that there really is something there - and that most attention is focused on project creation and sustainability and not in getting more t-shirts out to conferences your big org sponsors. There are times when you need to hold off on fully-open, e.g. oVirt. Getting partners lined up can be part of making something be there. In that case, we used a to-be-open list for planning, so in the end it was all transparent.

Generally, I think academic-sourced projects are more like small projects, except they have the chance to tap in to a Big Machine that can promote, support, budget, hold conferences, etc. The biggest risk is graduation - more people leave academia than stay. Companies don't deal with that, so they can heal quickly when one or two paid project roles need replacement. So it helps if the project is sponsored by a tenured professor, or that it can transition beyond the academic walls when the contributors graduate/move on.

Governments around the world are different, so it's hard to generalize. For the US gov't there is the situation where work should be in the public domain (which a FOSS license is not), and they often work with contractors who come from traditional software development backgrounds. So a gov't has to make sure that the call for proposals includes that the resulting software is FOSS, etc. One experience I had was with SELinux, where we worked with the NSA. They needed help in creating a real, sustainable project; getting sources upstreamed to the kernel; growing the ecosystem beyond a few small contractors; and making it possible for vendors to create COTS Products. The irony is, gov't offices have to procure solutions usually, so if they want to *use* FOSS, it has to be in a product. They can write FOSS, but have to be aware of security, privacy, and issues that a private org thinks differently about.

Considizations for project size and origin (cont'd)

- Sponsored-by-large-org projects are different because …
  - Stigma & worries about abuse of power.
  - Governance is highly important.
  - Be careful with marketing machine.
- Academic-sourced projects are different because ...
  - Small with a potential Big Machine behind it.
  - Graduation risk – continuity needed.

Stigma of the large org. People worry about abuses of power. Quality governance is super-important - people have to see how it's enshrined that the large-org isn't going to abuse power. It's easy to have the marketing machine get involved, which can help boost attention. Be very careful that the initial steps are done - that there really is something there - and that most attention is focused on project creation and sustainability and not in getting more t-shirts out to conferences your big org sponsors. There are times when you need to hold off on fully-open, e.g. oVirt. Getting partners lined up can be part of making something be there. In that case, we used a to-be-open list for planning, so in the end it was all transparent.

Generally, I think academic-sourced projects are more like small projects, except they have the chance to tap in to a Big Machine that can promote, support, budget, hold conferences, etc. The biggest risk is graduation - more people leave academia than stay. Companies don't deal with that, so they can heal quickly when one or two paid project roles need replacement. So it helps if the project is sponsored by a tenured professor, or that it can transition beyond the academic walls when the contributors graduate/move on.

Governments around the world are different, so it's hard to generalize. For the US gov't there is the situation where work should be in the public domain (which a FOSS license is not), and they often work with contractors who come from traditional software development backgrounds. So a gov't has to make sure that the call for proposals includes that the resulting software is FOSS, etc. One experience I had was with SELinux, where we worked with the NSA. They needed help in creating a real, sustainable project; getting sources upstreamed to the kernel; growing the ecosystem beyond a few small contractors; and making it possible for vendors to create COTS Products. The irony is, gov't offices have to procure solutions usually, so if they want to *use* FOSS, it has to be in a product. They can write FOSS, but have to be aware of security, privacy, and issues that a private org thinks differently about.

Considerations for project size and origin (cont'd)

- Sponsored-by-large-org projects are different because …
  - Stigma & worries about abuse of power.
  - Governance is highly important.
  - Be careful with marketing machine.
- Academic-sourced projects are different because ...
  - Small with a potential Big Machine behind it.
  - Graduation risk – continuity needed.
- Government projects are different because …
  - Must require contractors to open source code.
  - Procurement rules require COTS.
  - Privacy, security, other issues require diff thinking.

Stigma of the large org. People worry about abuses of power. Quality governance is super-important - people have to see how it's enshrined that the large-org isn't going to abuse power. It's easy to have the marketing machine get involved, which can help boost attention. Be very careful that the initial steps are done - that there really is something there - and that most attention is focused on project creation and sustainability and not in getting more t-shirts out to conferences your big org sponsors. There are times when you need to hold off on fully-open, e.g. oVirt. Getting partners lined up can be part of making something be there. In that case, we used a to-be-open list for planning, so in the end it was all transparent.
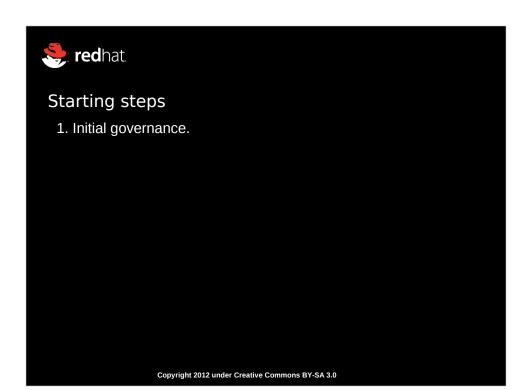
Generally, I think academic-sourced projects are more like small projects, except they have the chance to tap in to a Big Machine that can promote, support, budget, hold conferences, etc. The biggest risk is graduation - more people leave academia than stay. Companies don't deal with that, so they can heal quickly when one or two paid project roles need replacement. So it helps if the project is sponsored by a tenured professor, or that it can transition beyond the academic walls when the contributors graduate/move on.

Governments around the world are different, so it's hard to generalize. For the US gov't there is the situation where work should be in the public domain (which a FOSS license is not), and they often work with contractors who come from traditional software development backgrounds. So a gov't has to make sure that the call for proposals includes that the resulting software is FOSS, etc. One experience I had was with SELinux, where we worked with the NSA. They needed help in creating a real, sustainable project; getting sources upstreamed to the kernel; growing the ecosystem beyond a few small contractors; and making it possible for vendors to create COTS Products. The irony is, gov't offices have to procure solutions usually, so if they want to *use* FOSS, it has to be in a product. They can write FOSS, but have to be aware of security, privacy, and issues that a private org thinks differently about.

Starting steps in tl;dnr format

1. Initial governance.
2. Contribution policy.
3. External main project mailing list.
4. Source control for code and content.
5. Issue tracker is a general tool or method for the community to keep track of important issues (projects, problems, tasks) in a central way.
6. Wiki for community, collaborative documentation.
7. Weekly IRC meeting time.
8. Team planet/blog feed.
9. Open roadmap for the project on the wiki.
10. Simple open marketing plan, posted on project wiki, talked about on main mailing list – events, people to focus on, press, online seminars, etc.
11. Expose interesting and easier tasks – encourage peripheral participation.
12. Volunteer mentors wiki page.
13. How to participate and contribute page.
14. Community information page – including leadership and press contacts.
15. Participant and contributor improvements and needs page - wish list and roadmap for how things can/should improve for contributors and participants, over time.

Starting steps

1. Initial governance.

redhat.

Starting steps

1. Initial governance.
2. Contribution policy.

Starting steps

1. Initial governance.
2. Contribution policy.
3. External main project mailing list.

Starting steps

1. Initial governance.
2. Contribution policy.
3. External main project mailing list.
4. Source control for code and content.

redhat.

Starting steps

1. Initial governance.
2. Contribution policy.
3. External main project mailing list.
4. Source control for code and content.
5. Issue tracker is a general tool or method for the community to keep track of important issues (projects, problems, tasks) in a central way.

Starting steps

1. Initial governance.
2. Contribution policy.
3. External main project mailing list.
4. Source control for code and content.
5. Issue tracker is a general tool or method for the community to keep track of important issues (projects, problems, tasks) in a central way.
6. Wiki for community, collaborative documentation.

Starting steps

1. Initial governance.
2. Contribution policy.
3. External main project mailing list.
4. Source control for code and content.
5. Issue tracker is a general tool or method for the community to keep track of important issues (projects, problems, tasks) in a central way.
6. Wiki for community, collaborative documentation.
7. Weekly IRC meeting time.

redhat

Starting steps

1. Initial governance.
2. Contribution policy.
3. External main project mailing list.
4. Source control for code and content.
5. Issue tracker is a general tool or method for the community to keep track of important issues (projects, problems, tasks) in a central way.
6. Wiki for community, collaborative documentation.
7. Weekly IRC meeting time.
8. Team planet/blog feed.

redhat.

Starting steps

1. Initial governance.
2. Contribution policy.
3. External main project mailing list.
4. Source control for code and content.
5. Issue tracker is a general tool or method for the community to keep track of important issues (projects, problems, tasks) in a central way.
6. Wiki for community, collaborative documentation.
7. Weekly IRC meeting time.
8. Team planet/blog feed.
9. Open roadmap for the project on the wiki.

redhat

Starting steps (cont'd)

10. Simple open marketing plan, posted on project wiki, talked about on main mailing list – events, people to focus on, press, online seminars, etc.

Starting steps (cont'd)

10. Simple open marketing plan, posted on project wiki, talked about on main mailing list – events, people to focus on, press, online seminars, etc.
11. Expose interesting and easier tasks – encourage peripheral participation.

Starting steps (cont'd)

10. Simple open marketing plan, posted on project wiki, talked about on main mailing list – events, people to focus on, press, online seminars, etc.
11. Expose interesting and easier tasks – encourage peripheral participation.
12. Volunteer mentors wiki page.

redhat

Starting steps (cont'd)

10. Simple open marketing plan, posted on project wiki, talked about on main mailing list – events, people to focus on, press, online seminars, etc.
11. Expose interesting and easier tasks – encourage peripheral participation.
12. Volunteer mentors wiki page.
13. How to participate and contribute page.

Starting steps (cont'd)

10. Simple open marketing plan, posted on project wiki, talked about on main mailing list – events, people to focus on, press, online seminars, etc.
11. Expose interesting and easier tasks – encourage peripheral participation.
12. Volunteer mentors wiki page.
13. How to participate and contribute page.
14. Community information page – including leadership and press contacts.

Starting steps (cont'd)

10. Simple open marketing plan, posted on project wiki, talked about on main mailing list – events, people to focus on, press, online seminars, etc.
11. Expose interesting and easier tasks – encourage peripheral participation.
12. Volunteer mentors wiki page.
13. How to participate and contribute page.
14. Community information page – including leadership and press contacts.
15. Participant and contributor improvements and needs page - wish list and roadmap for how things can/should improve for contributors and participants, over time.

Sustaining steps in tl;dnr format

1. Get out of the way & let it thrive.
2. Improve your infrastructure - focus on enabling contributions.
3. Remember what you learned in Kindergarten - be nice, share.
4. Focus on legitimate peripheral participation.
5. All discussions and decisions must default to OPEN.
6. Use version control for content & code.
7. Choose open tools that can be extended.
8. Focus on keeping the community healthy.
9. Keep governance real and active.
10. Deal with poison.
11. Ongoing communicator(s) are needed.
12. Consensus over voting.
13. Release early, release often.
14. Use a predictable schedule, stick to it.

Sustaining steps

1. Get out of the way & let it thrive.

redhat

Sustaining steps

1. Get out of the way & let it thrive.
2. Improve your infrastructure - focus on enabling contributions.

redhat.

Sustaining steps

1. Get out of the way & let it thrive.
2. Improve your infrastructure - focus on enabling contributions.
3. Remember what you learned in Kindergarten - be nice, share.

Sustaining steps

1. Get out of the way & let it thrive.
2. Improve your infrastructure - focus on enabling contributions.
3. Remember what you learned in Kindergarten - be nice, share.
4. Focus on legitimate peripheral participation.

Sustaining steps

1. Get out of the way & let it thrive.
2. Improve your infrastructure - focus on enabling contributions.
3. Remember what you learned in Kindergarten - be nice, share.
4. Focus on legitimate peripheral participation.
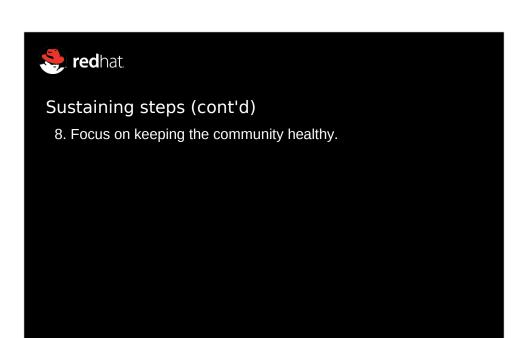5. All discussions and decisions must default to OPEN.

redhat.

Sustaining steps

1. Get out of the way & let it thrive.
2. Improve your infrastructure - focus on enabling contributions.
3. Remember what you learned in Kindergarten - be nice, share.
4. Focus on legitimate peripheral participation.
5. All discussions and decisions must default to OPEN.
6. Use version control for content & code.

redhat.

Sustaining steps

1. Get out of the way & let it thrive.
2. Improve your infrastructure - focus on enabling contributions.
3. Remember what you learned in Kindergarten - be nice, share.
4. Focus on legitimate peripheral participation.
5. All discussions and decisions must default to OPEN.
6. Use version control for content & code.
7. Choose open tools that can be extended.

Sustaining steps (cont'd)

8. Focus on keeping the community healthy.

redhat.

Sustaining steps (cont'd)

8. Focus on keeping the community healthy.

9. Keep governance real and active.

Sustaining steps (cont'd)

8. Focus on keeping the community healthy.
9. Keep governance real and active.
10. Deal with poison.

Sustaining steps (cont'd)

8. Focus on keeping the community healthy.
9. Keep governance real and active.
10. Deal with poison.
11. Ongoing communicator(s) are needed.

Sustaining steps (cont'd)

8. Focus on keeping the community healthy.
9. Keep governance real and active.
10. Deal with poison.
11. Ongoing communicator(s) are needed.
12. Consensus over voting.

Sustaining steps (cont'd)

8. Focus on keeping the community healthy.
9. Keep governance real and active.
10. Deal with poison.
11. Ongoing communicator(s) are needed.
12. Consensus over voting.
13. Release early, release often.

Sustaining steps (cont'd)

8. Focus on keeping the community healthy.

9. Keep governance real and active.

10. Deal with poison.

11. Ongoing communicator(s) are needed.

12. Consensus over voting.

13. Release early, release often.

14. Use a predictable schedule, stick to it.

redhat.

Growth steps

1. Evolve with community.

redhat.

Growth steps

1. Evolve with community.
2. Change leadership regularly, democratically-by-merit.

Growth steps

1. Evolve with community.
2. Change leadership regularly, democratically-by-merit.
3. Be students as well as teacher/mentors.

redhat.

Growth steps

1. Evolve with community.
2. Change leadership regularly, democratically-by-merit.
3. Be students as well as teacher/mentors.
4. Drive down barriers.

Growth steps

1. Evolve with community.
2. Change leadership regularly, democratically-by-merit.
3. Be students as well as teacher/mentors.
4. Drive down barriers.
5. Embrace failure.

redhat

Growth steps

1. Evolve with community.
2. Change leadership regularly, democratically-by-merit.
3. Be students as well as teacher/mentors.
4. Drive down barriers.
5. Embrace failure.
6. ... and more!
   - This is an example of where TheOpenSourceWay.org is incomplete and needs more, more, more.

redhat.

Let's talk about the risks and rewards of growth ...

- Risks:
  - Growing too fast, too big, too diverse.
  - Loss of focus.
  - Dilution of vision.
  - Flamewars & bikeshedding.

redhat.

Let's talk about the risks and rewards of growth ...

- Risks:
  - Growing too fast, too big, too diverse.
  - Loss of focus.
  - Dilution of vision.
  - Flamewars & bikeshedding.
- Rewards:
  - Take work off peoples' shoulders.
  - Larger, more creative innovation stream.
  - Greater reach locally and globally.

redhat

Let's talk about the risks and rewards of growth ...

- Risks:
  - Growing too fast, too big, too diverse.
  - Loss of focus.
  - Dilution of vision.
  - Flamewars & bikeshedding.
- Rewards:
  - Take work off peoples' shoulders.
  - Larger, more creative innovation stream.
  - Greater reach locally and globally.
- What is the "right size" for your project?

redhat.

Conclusions?

Questions?

Discussion?

redhat

Anyone here need some help today?