



FLOSS

MANUALS



Table of Contents

Introduction.....	1
The Aims.....	3
Book.....	3
Content.....	3
Community.....	3
The Process.....	5
Obtaining a Commitment.....	5
Pre-planning.....	5
The sprint.....	5
Post-sprint tasks.....	6
Communities.....	7
Meeting in One Place.....	7
Remote contributions.....	9
Pre-Planning.....	10
Obtaining funds.....	11
Determining topic and scope.....	11
Topic and audience.....	11
Scope.....	12
Writing the outline.....	12
How much detail?.....	12
Index.....	13
Establishing tone and style.....	13
Finding authors.....	14
Reuse Existing Content.....	14
Handling logistics.....	14
Anticipating post-sprint tasks.....	15
Roles.....	16
Sprint Facilitator.....	16
Local Host.....	16
FLOSS Manuals Expert.....	16
Manual Maintainer.....	16
Participants.....	17
External Editors / Reviewers.....	17
Designers/ Illustrators.....	17
Time Line and Schedule.....	18
Logistics and Budget.....	19
Travel.....	19
Accommodations.....	19
Food.....	19
Remuneration.....	20
Writing Resources.....	20
Venue.....	21
Tools.....	25
Collaborative authoring and publishing.....	25
Status indicators.....	25

Table of Contents

Tools	
Permissions.....	25
Remote Contributions.....	26
Inline Chat.....	26
IRC.....	27
Email Notifications.....	27
Editing Notifications.....	27
Tips for sprint writers.....	29
Adams Thoughts.....	30
Writers vs Contributors.....	30
Team Development.....	30
Books?.....	30
Publisher?.....	30
Living Texts.....	30
Food and Venue.....	31
Creating the Index.....	31
Combining another Event.....	32
Narrative Flow.....	33
Getting People Started.....	33
Writing as Conversation.....	33
Disruptive Conversation.....	34
Gaining Trust.....	34
The Shape of the Week.....	34
Pairing it up.....	35
Structural jams.....	35
Harmonising Flow.....	36
Tone.....	37
Example 1.....	37
Example 2.....	37
Serendipity.....	37
Follow Up.....	38
Inkscape Book Sprint.....	39
Funding.....	39
Invite List.....	39
Location.....	39
Lasting Outcomes.....	40
XO Sprint.....	41
Funding.....	41
Platform Development.....	41
Invite list.....	42
Location and date selection.....	42
Outputs.....	42
Perspective from a participant.....	45
FLOSSify.....	47
Platform Development.....	47
The Sprint.....	48

Table of Contents

Command Line Manual.....	51
Platform Development.....	51
Pre-planning.....	52
Set-up.....	53
Process.....	54
Post-sprint tasks.....	54
Evaluation.....	55
Outcomes.....	55
CiviCRM.....	57
The CiviCRM project and community.....	57
Platform Development.....	57
Organization.....	58
Controversy over outline.....	59
Sprint-week editing.....	59
Fun.....	60
Follow-up.....	60
Feedback.....	60
Open Translation Tools Sprint.....	63
The Venue.....	63
The Sprint.....	64
Book Two.....	65
Facilitator Contributor.....	65
Printing Out.....	65
License.....	67
Authors.....	68
General Public License.....	71

Introduction

This book describes how to carry out collaborative authoring in a short time with the express goal of having a publishable book at the end. The Book Sprint concept was devised by Tomas Krag. Tomas conceived of book production as a collaborative activity involving substantial donations of volunteer time.

Tomas pioneered the development of the Book Sprint as a 3-4 month production cycle, while Adam Hyde, founder of FLOSS Manuals, was keen to continue with the idea of an "extreme book sprint," which compressed the authoring and production of a print-ready book into a week-long process.

During the first year of the Book Sprint concept, Adam and FLOSS Manuals experimented with several models of sprint.



Sugar Labs and One Laptop per Child Book Sprint, August 2008

Book sprints have generally taken place in areas of computer technology, where documentation has traditionally been tarred as the most irksome and unpleasant of tasks. Book sprints prove that nothing could be farther from the truth. They make writing fun by:

- Identifying key areas where people need more information, and thus providing an impetus to do something valuable for the community.
- Elevating the writing process from an isolated to a group activity, making it a fulfilling personal experience and offering immediate feedback as a reward for one's effort.
- Giving participants a chance to focus intensely on the problem and process of producing helpful information, which allows them to appreciate the pleasures of the task and the subtleties of the accomplishment.

The result of the sprint is a book meeting the specifications of the community that began the project--in other words, it meets known needs.

Because book sprints involve open contributions (people can contribute remotely through a wiki as well as by joining the sprint physically) the process is probably most suited to books offered free online. Indeed, the goal

of FLOSS Manuals embodies this freedom in a two-fold manner: it makes the resulting books free online, and focuses its efforts on free software.



FLOSSify - Digital Foundations Book Sprint, February 2009

The Aims

The aims of a Book Sprint can be broken down to the following categories:

Book

A complete book motivates the process and helps justify the effort. A book is also a fantastic promotional and educational device for the free softwares written about.

Content

Rapid development of content helps a community quickly 'catch up' to their documentation needs.

Community

Mobilizing a community to produce its own documentation helps to forge bonds and build pride in other efforts by the community and better ensures the ongoing development and maintenance of the content.

This chapter discusses each of those aims, and includes some reasons for writing a book in a sprint instead of using more conventional methods.

Book

Traditional book production time lines are normally measured in months and years. Book Sprints produce comparable content in a much shorter amount of time. Using Book Sprints and print on demand technologies, the time scale from zero to published book is measured in days and weeks.

As an example, the FLOSS Manuals Circumvention Book Sprint was organised in two weeks and written in one. The sprint brought eight people together from around the world. We started work on the text at 9:00 AM Monday and finished with a beer on Friday at 6:00 PM. At that moment, with the click of a button, we generated the book-ready source files and uploaded them to the print-on-demand service.

Having a finished book was a wonderful way to wrap up an intense and pleasurable project, knowing that anyone could immediately buy the well laid out paper version of the manual we had just written. It also gave us something to blog about and something to tell our friends about. It was, all in all, an astonishing accomplishment--but not something that required superhuman talent or effort.

Content

While it is very motivating to have a book at the end, it is not actually the sprints' primary goal. The primary goal is to generate content.

A book sprint generally stems from the need to play "catch up" with documentation. There is so much work to be done in the world of free software documentation that there is always a need to produce more material, and to produce it rapidly.

Community

Book sprints are also wonderful for building communities around the topic. If you work intensely with people that are passionate about the same things you are, you naturally form friendships and the group develops the characteristics of a community. It is hard to convey the effect of sitting across the table with people for four or eight hours at a time, sharing ideas and sometimes arguing, eating and drinking with them, and talking about life.

Some sprints develop small, niche-orientated communities, which are nonetheless satisfying. Others (such as the OLPC and Sugar manuals produced by FLOSS Manuals, and the book on circumventing censorship) have

the potential to motivate thousands of people.

Our intention with each book sprint is to get the participants to know each other and continue to work together, maintaining and extending the material, as a direct result of the friendships they form at the sprint.

A book can be maintained by an individual or a small group and still be valuable. But over the long haul, individuals lose steam and small groups tend to disperse. The great thing is that community-building extends far beyond the six to ten people who meet for a sprint. With established communities (such as a free software project) much larger subsets of that community might participate in the planning that leads up to the event, and some sprints have a larger set of remote contributors (logging in at all times of the day and night over the FLOSS Manuals web interface) than the set of contributors physically present. It helps everyone interested in the project, where ever they are in the world, feel that they have something productive to contribute.

Although it is not always necessary, some sprints are followed by a week or two of additional activity to polish and reorganize the material. At this point people work on the content remotely, and sometimes add substantial content. To harness this energy and encourage the ongoing development of the content, it is good to have someone dedicated to the role of maintainer. Book sprints can start the ball rolling, but as everyone knows, there is no such thing as a perpetual motion engine. You always need to put energy in to keep the engine ticking.

The Process

A book sprint is a surprisingly light-weight process. Part of this stems from the use of the FLOSS Manuals platform. The FLOSS Manuals website allows us to build a book sprint around it with very little overhead, letting people concentrate on the important tasks--notably the actual writing. However, providing the technology isn't the entire story. Book Sprints are an evolving Social Methodology, and the FLOSS Manuals website is the best platform for using in this context.

What should be acknowledged however, is that the FLOSS Manuals platform is at the time of writing, only committed to manuals about free software. The platform is really key, as it makes the process easy, fast, and collaborative. Previous sprints done without this platform have battled with their tools whereas we have a very good platform which helps the process. The FLOSS Manuals foundation is currently working on the development of a new content-agnostic collaborative writing and book sprint platform. Stay tuned.

Obtaining a Commitment

First, someone has to decide that there needs to a Book Sprint on a particular subject.

At this point the project leader/initiator needs to decide:

- Can you use the FLOSS Manuals system (it can only be used for documenting Free Software)?
- Is the outcome worth the commitment? A book sprint usually produces a text of about 160 to 250 pages. This is not enough to fully cover some project's documentation needs. In some cases the leaders of the project must be prepared to choose a narrow topic during the pre-planning phase and avoid 'mission creep'. They must agree to feel satisfied by the outcome of the sprint.
- Who will participate, and who will take responsibility? If the project has certain recognized leaders (who may or may not hold official positions) their moral support is very useful for a project like this one. Support from such leaders will ensure that other people will take the project seriously and make the time to participate. However, you can also choose to document a software where there is no involvement from any of the 'official' team. Any one can choose to write documentation, and there is not enough of it in the world - why limit who can do it?
- How much time and money can be committed? Costs can be kept low in a variety of ways, but the organization should be prepared to spend some money.
- Who can maintain the document? It's demoralizing to produce a great book and have it become useless after a year or so because no one keeps it up to date.

Pre-planning

This takes two to six weeks. It usually involves a few people intensively (sometimes up to about 10 people) but can also draw in ideas from a larger community.

The sprint

This forms the topic of the bulk of this book. Although we provide as much guidance as we can, you should recognize that:

- Each sprint is unique. A leader needs enormous flexibility, along with the ability to listen to participants and find quick solutions.
- Book sprints are still at a young stage. Although they have all been considered successes, we are always experimenting with ways to make them better.
- You never really know what a people-based process is like until you've done it--and perhaps done it a few times.

- Maintaining a positive atmosphere and letting people go home feeling good is more important than doing it "right." Be happy with whatever outcome you get--it is very likely to be better than anyone expected anyway!

Post-sprint tasks

Although a sprint is not the chaotic, slapdash effort many people would imagine, the resulting book can always be improved. There might be gaps, as well as redundant material written by two different people who did not realize they were duplicating each other's work. The book might have inconsistencies in terminology, style, and point of view that should be harmonized to make it flow better. And sometimes sections need to be rearranged.

If this occurs then usually the participants of the sprint have the energy to make these fixes in the week or two following the sprint, and their enthusiasm should be harnessed to do this.

Over the long range, the project should appoint a maintainer who checks the book every couple months to see whether something is out of date or whether it would be useful to add new material. This maintainer can draw on sprint participants and others to add material. At some point, it may be worth undertaking a major revision in a new sprint. (FLOSS Manuals hasn't reached that stage on any book yet.)

Communities

Because documentation must be a living, breathing entity, it is usually associated with a community of practitioners. Many wonderful books are written by a single book, and even some book sprints involve a small, self-contained group rather than a fluid community. But communities can do a lot for books, for two main reasons.

First, if the content is linked to an existing community, the authors need to hear from that community what topics to cover, or the book the authors worked so hard on may end up proving irrelevant. (More likely, it will still be a useful book, but won't meet the most pressing needs of the community.)

Second, it is not wise to create documentation about free software or related topics without a strategy to maintain and update the content. Given the "release early release often" strategy so common in free software, the pace of the change can be dizzying.

Therefore, we recommend that every major documentation project reach out to the people who care about the project, both among the developers and those using the software. Communities can:

- Explain the topics where they need help (or, out of their experience, where other people treading the same path will need help)
- Contribute to the documentation as authors, editors, and reviewers
- Offer support and thanks to the authors
- Produce maintainers on an ongoing basis

So our mantra for the book sprints is :

"Written in a week. Maintained by a community."

Meeting in One Place

As David Garcia has said, "Face to face is the broadest bandwidth." We firmly believe in this mantra. When applied to book sprints, it equates to getting people together in one room to enable rich communication. Moreover, this communication and the friendships formed will help create a micro-community around the subject.

By providing face-to-face time writing in a book sprint we hope to secure both goals. When you have to share the same room you can talk about dense subjects more easily than you can online. You also get to know the people involved much better, which assists in understanding the points they might be trying to make in their writing.

For this reason it is also important to do things as a group during the book sprint week that do not involve writing.



Have a barbeque together; go on walks; see what the surroundings have to offer. It gives everybody a bit of a break so they can press their own "refresh" button and prepare them for the next period of writing. But beyond that, it fosters the ongoing community component of the book sprint's goals.



Also, if you have fun, people will want to do it again.



Remote contributions

Having said that, the face-to-face nature of community building is not absolutely necessary. We have built the FLOSS Manuals tool set so that remote contributions to a book sprint can work well. We have added some notification tools and chat systems so that there can be communication between writers online. These tools have actually proven so convenient that they are sometimes used even by those sharing the same room to talk to each other.

It is also important to note that FLOSS Manuals does a lot of work outside book sprints. The communication and collaboration tools we are building serve just as well for individuals or geographically separated groups to contribute to any of the manuals. This material, for instance, was written by many people without gathering for a sprint.

Pre-Planning

Although we have scheduled book sprints as little as three weeks after the decision to hold them, and have squeezed the necessary pre-planning into less than 2 weeks, we find that it's a good idea to allow at least a month for pre-planning. Remember that someone has to find a venue, that some people may have to make travel arrangements, and that the proposed contributors need to be polled for topics.

The Free Software Foundation and FLOSS Manuals planned a book (Introduction to the GNU/Linux Command Line) with very little time for pre-planning. But there were several mitigating factors that made it possible to pull the sprint together in that short time:

- It was held concurrently with their annual conference, on the same grounds. This meant that it was easy to find space, the book sprint participants could mooch off the main conference for food, and many conference participants could take a few hours off from the conference to join the sprint.
- The topic (the Bash shell and related command line issues) was familiar to the participants. This meant that an outline could be developed quickly (the initial version was created by one participant in about one hour).
- The prestige and publicity strengths of the Free Software Foundation helped recruit excellent writers (most of them remote). As a side effect, this allowed the authors to produce a sizeable book--over 200 pages--in just two days (note : FLOSS Manuals books use a non-standard page size, 200 pages in this format equates to about 150 A4 pages).

One or more people need to be in charge of planning the event. The tasks include:

Obtaining funds

The exact amount of money required will depend on the choices made for venue (and can be kept fairly low), but the source should be identified early in the planning process.

Determining topic and scope

Although an organization may make a top-down decision that it needs a manual on a particular topic, it's a good idea to hold some discussions beforehand (over email, teleconferences, or on a wiki) to find out from the grass roots what topics need to be covered.

Writing the outline

Some sprints have started with detailed outlines (down to one or two levels of heading in each chapter), whereas others have started with a loose grab-bag of topics. We are still experimenting with finding the right balance.

Establishing tone and style

This is a very subjective task. You want to provide guidelines so you approach your audience in the same manner throughout the book, while leaving scope for each author to have his or her own voice.

Finding authors

Luckily, within any organization that commits to running a sprint usually there are some highly motivated and accomplished members who can be persuaded to devote a week to the project. Otherwise you need to start researching and finding the people out there that know a lot about the topic. A number of people may be recruited, partly based on their ability to travel and devote the time.

Handling logistics

This is a large area of effort that covers finding a venue, providing resources, and arranging for food. Once you know the geographical area where the sprint should be held, someone who lives there should take on this task and find helpers if necessary.

Anticipating post-sprint tasks

It's not too early to talk about maintenance, although the energy generated by the sprint itself will help recruit maintainers.

Obtaining funds

Where do you get the funds? We have worked with partners that have sourced funds from NGOs, a little from Google's Summer of Code Initiative, and we have also done some fund raising ourselves. It's hard to give any general advice on this as we are young at fund raising ourselves.

Determining topic and scope

The two big questions to answer before a sprint are:

Topic and audience

What problem are you trying to solve? This also necessarily involves discussion of audience (the goals and background of your intended readers.

Scope

This could be defined as "Knowing when you're finished." It involves an analysis of the topic to determine what you need to say, and some prioritization to give the sprint achievable goals.

Topic and audience

The topic may be determined by the person who conceives of the sprint. For instance, the first sprint at FLOSS Manuals started with a statement as simple as "this manual is going to be about Audacity."

Still, it's important to have a strong concept of whom you're writing for and why. For instance, to plan a manual about Audacity (an audio editing tool), you have to ask whether your audience understands basic audio-related technical topics, such as sound quality and the characteristics of audio digital files. You also have to ask what you want to help them accomplish. Simple cutting? Multitrack editing?

A community can often contribute a lot to the choice of topic. Sometimes, project leaders may have heard requests for a particular manual and may know what they want to cover without further consultation with the community. But it can be surprising what they discover if they throw a preliminary chapter list out for review.

Probably the most extensive discussion of topic for a FLOSS Manuals book came for CiviCRM. It has a large community spread over at least three continents, and about a dozen people participated in a mailing list to plan their sprint. When it became clear that many valid and useful topics were being floated by various leaders, we held a teleconference about ten days before the event to pare away topics and decide what was really the most pressing issue they needed to cover.

Some questions to ask about your audience might be:

- What (if any) are the job titles they hold?
- What organizations or groups (if any) do they belong to?
- Are they amateurs with limited knowledge or industry professionals in the workforce? Some--such as students--may occupy an intermediate position.
- What other publications might they read, or what subject matter should they already be familiar with?
- What is their general level of computer literacy?
- What operating systems are they most commonly going to use?
- What preconceived notions might you have to dispel?

It is also always a good idea to have one person from that target audience in the sprint. For example, if the target audience is "newbies", invite a newbie to review each section. Instead of trying to imagine what level to pitch the material at, it is much more valuable to have a member of your target audience in the room to look the experts in the eye and say "I don't understand what that means". The experts may then have to recalibrate their tone. This is not to say that the target audience member is always right, but the experts have to justify

their position when challenged in this manner, and that is going to lead to better texts.

Scope

Once a main topic has been determined, you need to decide what subtopics to cover. Each book needs to have a beginning--which represents what your audience already knows--and an end.

As an example of the relationship between scope and audience, the Command Line manual started with such rock-bottom basics as how to find the Terminal program on your graphical desktop. The editor who wrote the outline (Andy) realized that a simple task like finding the program that lets you run a command line could prove a stumbling block to our readers. Another introductory topic was to explain what sort of possibilities are opened up to you if you master the command line, so as to motivate people to read the manual.

To set the manual's scope, start brainstorming as far as possible before the sprint. Try to create an atmosphere of acceptance, so people's suggestions aren't rejected hastily. This pre-planning should also help you identify potential participants in the book sprint: for instance, the technical experts you need, and representatives of the target audience.

However, there are two constraints you must respect:

- A sprint must ultimately have a limited scope. Because most books come out less than 250 pages in length, some worthy topics will have to be left out. At FLOSS Manuals sometimes we use a benevolent dictator approach, where the Sprint Master determines the outline shortly before the sprint. For the CiviCRM sprint, that approach was rejected, so we used another approach that worked well: the team that assembled to write the book decided on the scope during the first morning of the sprint.
- You can include only the topics that the participants are competent to write about, and interested in writing about. Of course, you can seek other subject matter experts to fill in sections after the sprint.

Knowing how much you can accomplish in the short time you have during a sprint is not exactly a science. An experienced Sprint Master can help identify what's feasible. An experienced Sprint Master will always tell you anything is possible (especially if you have a lot of remote contributions)...but some things are more possible than others of course...

Keep in close touch with the project leaders who have funded the sprint during the pre-planning, and make sure they will be satisfied with the scope. The Sprint Master must get them to understand that there are limits to what can be accomplished in a week and within 160 to 250 pages. They also must understand that unexpected considerations may come up during the writing, and that they must trust the sprinters to collectively make the right decision about scope.

Writing the outline

Some FLOSS Manuals books start with a chapter devoted to an outline. It's useful to ask people to edit this. The outline not only reveals the scope of the book--by listing all the topics--but puts the topics in logical order. Other times the outline has been generated on the spot by the participants by using the Index builder.

How much detail?

At FLOSS Manuals, we're still working out the level of detail that an outline should have as we go into the sprint. Just as the sprint as a whole must strike a balance between being productive and having fun, the level of detail in an outline is a balance between giving people a guide about what to write and leaving them enough room for the spontaneity and creativity that produces the best writing.

A detailed outline has one potential benefit: authors who write up advanced material can take it for

granted that readers will get the necessary background earlier in the manual.

For instance, in the Command Line manual, advanced chapters on scripting could be written quickly under the assumption that readers already knew the building blocks of scripts (variables, control structures, and so on). It would severely hamper the authors if they had to worry about whether their readers knew these building blocks. They might have wasted time writing up this material and exhausted the time they meant to spend on useful advanced topics. On the other hand, they might have left out important information and no one might have realized until afterward that a big information gap was left.

Still, it's a waste for the outline to include details that the authors will later ignore. Each author has a unique vision, and sometimes a structure that looks great in the abstract turns out to be inferior when it's fleshed out into a chapter.

Two potential hazards when building outlines are:

- the outline might end up being too broad if too much time is spent on it - this can build up unrealistic expectations on how much content will be written
- if the participants of the sprint are not the same as the 'outline builders' there will be a critical disconnect between these two groups that will need to be fixed

Both these issues have been solved in previous sprints (eg. Inkscape and CiviCRM sprints) by re-building the Index with all book sprint participants involved as soon as everyone arrives on site.

Index

In FLOSS Manuals, the index is the set of web pages that make up the book. (This is different from the kind of index you find at the back of a print book.) When people visit the main page for the book on FLOSS Manuals, the index is the list of topics they see. The index also becomes the chapters in the book.

A Sprint Master or other team members often create an index during the pre-planning phase. But most sprinters end up changing the index radically, adding and removing sections as they encounter new needs. Everyone has to keep a very flexible attitude to it as the sprint goes on. As one sprint participant said, "It's about the constantly redefining what complete is."

Establishing tone and style

A book can be frustrating if it switches tone in the middle. One author may write in a jazzy, loose style, such as "Don't panic--we'll reveal the wizardry in a minute," while another might write in a more formal style, saying "The following example is complex, but will be understandable by the time you finish the chapter." Each style is legitimate and useful, but the reader will feel queasy if the tone makes a big swing from one style to another.

On a larger scale, you want authors to make similar decisions about when to introduce background material, how to intersperse examples with explanations, and other issues of flow.

Each FLOSS Manuals book includes a chapter on Writing Conventions. However we don't believe it's a good idea to push this on writers. The conventions are really there if someone asks for them. Otherwise we keep them out of the way and then tell them to just start writing.

However, if you do want to set some kind of style, perhaps the best way to convey tone and style is by example. Before the Command Line sprint, the editor Andy wrote a chapter in the style he wanted for the book. He also provided strongly worded advice about the style we were seeking in the Writing Conventions.

Although we don't know how many sprinters read the sample chapter or the Writing Conventions, we found that nearly every contribution adhered to the guidelines concerning pace and the approach to the reader, a feat all the more remarkable considering the large number of contributors.

Finding authors

We have conducted a fair number of Book Sprints by now, and it always seems that we get the right people every time. The reason for this is because we allow those involved to decide what the manual will be about. If participants see that there is an area of documentation that needs to be covered, or they offer to cover something because they know about it, chances are good that they will write that material.

So issuing an open invitation is usually a successful starting point. However of course, there might be some people you think should absolutely be there. Then you need to make direct invitations.

Often, you probably won't get a choice. Seldom will all the people you want actually have time to come. Encourage your most valuable experts to participate remotely as time permits. This means playing a bit with the tools before the sprint, obtaining an account on FLOSS Manuals (if the sprint is using that site), logging in from time to time, and making themselves available over email or chat for questions.

Relationships among sprinters is also important, because a book sprint involves intense discussions under pressure. How people interact therefore plays a large role in what you produce. For this reason, the process is largely reliant on the chemistry between people, which is never easy to predict. The Sprint Master will play an important role in conjuring this chemistry, but chance also plays an important role.

Don't be too prescriptive - let a little chance bring something interesting and unexpected.

Reuse Existing Content

It is important to spend time finding as much existing content as possible. Re-use; re-use; re-use. Search the web and book stores for related content and write to each of the authors to ask whether you can re-use it. If it's a FLOSS Manuals book sprint, it's good to explain that this is a not-for-profit good faith exercise. This will help some potential contributors understand where we are coming from and they might feel more inclined to give copyright clearance. It is also important to ask whether the material can be used under the license you choose (FM uses the GPL).

If material is available and it is under a liberal license that permits free re-use, you don't legally have to ask for permission, but it's good ethics and fair treatment to write to the author and ask not their permission but their "blessing" to re-use the content. Many of those authors like to know where their content ends up and you might also find you have a new enthusiastic contributor come on board as the result of your communication. At the very least, they will probably tell other people about the project, and that's good word of mouth PR for you.

Getting existing material is important not just because it can contribute to the total content of the book but because it is motivating for the writers to see that some of the work has already been done before they start. In the end, the writers might not use the content but by then it has had its motivating effect.

Handling logistics

You need to plan of the dates, location, and participants in the Sprint before you get very far. If a group of participants is identified early in the pre-planning, they may pick the bests dates and location for them. Other times, a few project leaders have to pick a date and place and then see who is

available. Either way, you need to consider some basic issues. We'll cover some of these in more detail in the chapter on Logistics and Budget.

Venue

The choice should combine convenience for those attending, availability of space and resources, and attractiveness as a location. A section of this book is devoted to venue.

Travel

How will everyone get there? Will you find the funds for travel, or will the participants need to cover some or all of this themselves?

Accommodations

Where will everyone sleep? Will they pay for their own accommodations or will you find funds for it? Are the accommodations sufficiently close to the Book Sprint venue so that participants don't waste energy traveling every day. Do heavy snorers need their own room? Do you need to have a cooking space? The ideal venue has sleeping space for everyone, and cooking facilities.

Resources

This concerns both physical resources--such as tables, chairs, and a screen for presentations--and computing resources, especially high-bandwidth Internet access.

Food

Make sure the participants don't have to worry about where their next meal is coming from (and provide snacks too). Take into account health, pleasure, and diverse dietary needs.

Fun

Because you want participants to bond personally--and because they deserve a good time after their work each day--find extracurricular activities. These could be as simple as a walk around nice grounds if you're in the country, or as formal as a canal ride on tourist boats.

Anticipating post-sprint tasks

If you want the manual to grow you will need to ask yourself - who will facilitate this growth? You don't need to determine this before the sprint starts but it might be helpful. At most sprints identifying the person to take on the ongoing maintainer role is a natural product of the process.

Roles

A good book sprint needs someone who will take responsibility for each of the following roles. Naturally, there can be helpers for each role, but you want one person who ultimately will make sure the important tasks are redone. Often one person will take on many roles.

Sprint Facilitator

Also known as the Sprint Master, this is the person ultimately responsible for seeing to it that the sprint happens and runs smoothly. The sprint facilitator keeps a high-level vision of the project's goals, makes sure that things happen on time, and coordinates all parties. If an organization requested and funded the sprint, the sprint facilitator is the person ultimately responsible for delivering a book that satisfies this organization.

This job is more about social engineering than logistics. The Sprint Facilitator must be comfortable asking people to improve their game as much as being there to encourage and motivate everyone. The Sprint Facilitator does not need to be very familiar with the nuances of the content. It is better to find someone who is sensitive and good at listening, and additionally has good organizational skills, than someone who is weak in those skills but is an expert on the subject.

The Sprint Facilitator, if experienced, could potentially be the same person as the local host, although this is a big doubling-up of tasks. The more experienced the Sprint Facilitator is, the more productive the sprint will be, and there will also be a tangible positive difference in the quality of the content.

Local Host

Every book sprint needs someone who is on site beforehand--ideally someone who lives in the area and knows all its quirks--to handle the logistics of food, venue, Internet access, etc. This person makes sure that all the physical resources needed by sprinters are in place when they start, and is responsible for picking up last-minute items, bringing in the food, and taking people places on an on-the-fly, as-needed basis. This person may have to drive the sick writer to see a doctor or find the shopping center to buy that badly needed power adapter.

Beforehand, this person should be able to answer any questions from out-of-town visitors about travel, local transportation, and accommodations.

It is a mistake to think this person's job finishes as soon as everyone turns up: the local host or delegates will spend a great deal of time running around. Avoid asking one of your key writers to take on this logistical role; he or she will not be able to spend as much time writing as you hope.

FLOSS Manuals Expert

Although FLOSS Manuals is not a difficult system to learn quickly in a few hours, it is helpful to have someone familiar with the FLOSS Manuals toolset available during the Book Sprint hours to answer questions and offer troubleshooting advice if things go wrong. This person will preferably have a direct line to the FLOSS Manuals developers, or other experienced FM people.

Manual Maintainer

A manual hosted on FLOSS Manuals usually, although not always, is associated with someone who manages the maintenance of the book after the sprint. The maintainer often creates the style guide before the Book Sprint so that writers have it available immediately when they start writing. Typically the maintainer is one of

the writers. The person has an important role during the first few weeks after the sprint to do the clean-up that is inevitably required on the book. He or she should agree to keep returning to the manual to determine whether it needs updates, and to help find his or her own replacement when necessary.

Participants

The participants are the heart and soul of the Book Sprint, and without at least a few dedicated people a Book Sprint cannot take place. Sprint participants are going to write the content, and many edit each other's content as well. Some of them must have strong, pre-existing knowledge about the topic that the Sprint documents. Preferably (although it is not essential) they should know some simple HTML code and also be able to create and edit screenshots or diagrams as necessary.

External Editors / Reviewers

Another valuable role at a Book Sprint is someone who reads through and edits material, providing feedback when ever possible. Although writers can do this for each other (and remote participants can help as well), the book tends to come out more readable and coherent if work is monitored by someone outside the sprint room.

This person can use the same FLOSS Manuals interface to edit chapters when the contributors exit them, or can print out completed chapters and mark them up by hand (some Editors prefer this older editing method). If hard-copy is used, writers can enter the edits into the source files.

The external editor should be able to examine a chapter and the whole book for flow, make sure the logic of a chapter makes sense, and do low-level proofreading for spelling, punctuation, and grammar. Ideally, they can also judge the choices of language to ensure ease of translation later - but in reality this last point has never occurred due to the compressed time frame of a sprint. Having a remote editor can be quite useful, because he or she is not caught up in local dynamics and is less likely to suffer from groupthink.

Designers/ Illustrators

A book cover must be completed by the end of the week of the Sprint. Although the cover designer doesn't have to attend the Sprint, it's important for him or her to be in tune with the goals of the sprint and the style used by the authors in order to gather ideas for the design.

It's also valuable to have someone good at reducing ideas to icons. FLOSS Manuals already has icon sets, so the icon-maker doesn't have to be a designer as such, but must be able to manipulate SVG icons (we recommend using Inkscape) to form easily understandable diagrams.

The page layout is actually taken care of by the Objavi tool in FLOSS Manuals. Someone needs to go through the chapters, get rid of extraneous tasks, and look at the layout. Objavi takes care of the rest. Objavi does on-the-fly book formatting for uploading books to a print-on-demand service. The books are preformatted in the FLOSS Manuals design, but it is possible to alter the look and feel using the CSS window provided in Objavi.

Time Line and Schedule

Book Sprints can have different time lines based on several factors. If there is a lot of content already available and the intent is to organize it into a book, that may not take as long as starting from a blank screen. A sprint that has a lot of authors may also take less time. In general, we have found that a five day sprint to be most effective.

Here are some guidelines for a schedule we have used.

Sunday	10-20 minutes for training on FLOSS Manuals tool set, and 2-3 hours for outlining the entire book within the FLOSS Manuals tool. Use a projector for demonstrating FLOSS Manuals and for filling out the chapters and sections to be written in the upcoming week. Arrange for dinner together to let the writers get to know each other socially over food and drinks. Because, some people may be traveling long distances to attend, make sure the planning comes before the socialising. You can be sure socialising will take place later, but it is important everyone is present at the planning discussions.
Monday	A full day of writing from about 9:00 AM to about 6:00 PM, with a lunch break. Short review after or before dinner. Socialise at night.
Tuesday	Another full day of writing. Review at midday. Perhaps let people choose whether to hang out or write in the evening.
Wednesday	Another full day of writing. Freeze all chapters - no more chapters to be added after this point.
Thursday	The sprint within a sprint. Start at 9.00 AM and write until all the content is entered or you fall off your chairs with fatigue. All content must be completed by the end of the day.
Friday	A full day doing only cleanup on the content that is considered complete. Print the entire manual and proof it before getting it ready for print production by adding the cover. Plan some fun evening event as a reward.

If you have remote participants, you may want to schedule a daily conference call. In this call, all persons give their status and describe any blocks they encounter that are preventing progress on any part of the manual. This meeting should occur at the beginning of each day, with the understanding that sometimes that time might be shifted on any specific day.

Logistics and Budget

A Book Sprint can be done for very cheaply if all participants are local, you have a place to work in, and people volunteer. However, if you are bringing in experts from around the world or you wish to hire an experienced Sprint Master, you'll have more costs.

Still, the total cost of a fully funded international sprint is comparable to hiring a contract technical writer for a specific deliverable, yet the results from the collaborative authoring effort will be quicker, the process is a good team builder, it will help build a community around the content, may generate more interest in the project, and the quality we have seen so far is quite high.

Given that you're asking people to take a week away from jobs and family without compensation (probably), try to get enough funding to make sure it cost the participants nothing. In other word, pay for all travel, food, and accommodations. It's reasonable to ask them to use the lowest-cost travel options and find spare couches at local supporters' apartments. If the participants will need to pay for anything, make sure they know in advance. For instance, state clearly which meals are provided and which dates you have accommodations available, so they know whether they need to pay for anything at the two ends.

This chapter covers the items you will need to consider.

Travel

Try and cover everyone's travel if you can. Travel costs are usually larger than you anticipate. If someone flies in, for example, try to cover not only the flight but the costs of their travel to the airport from home and back again, and to the venue from the airport and back again.

Sometimes you may do a call for voluntary participation and find that some pay for their own travel to come from afar. This is fine, so long as they know what the deal is.

Accommodations

If you don't have a budget for a hotel (which can be expensive when you add up several people staying all week), ask around and find out who has a spare bed or couch. If you have a budget, consider hiring a vacation rental. These are cheaper than hiring multiple hotel rooms. A vacation rental is usually a fully equipped house with kitchen, etc., so you can also cook there and save some money on food instead of eating out.

The ideal book sprint is a house away from city distractions, with a good Internet connection, some space for people to escape to when they need some space, and enough beds so everyone can sleep there.

Food

Not only is it respectful to take care of participants food needs without charging them, but sprints go more efficiently if food just shows up when its needed and the participants don't have to waste time worrying about it. In addition to breakfast and lunch, provide snacks during the sprint.

Find out what their favourite foods are and go shopping. Make sure to explicitly ask what dietary restrictions they have (religious, political, and medical) and make sure you have something for everybody at every meal.

If you like to cook, you will have a fun week! If you don't like cooking, see if you can convince someone else to cook for the week.

If participants are responsible for their own meals in the evenings, be sure they know what they can expect to pay and where to find eating establishments at a variety of price levels. In a vacation rental, provide all meals on site (participants are usually happy to help cook when they're not working on the sprint).

Remuneration

If you have the budget for it, try to pay participants for contributing to the book. Some people stick to a strict creed of voluntarism, but others will find it easier to participate and be more motivated if you can reward them with a bit of money.

If you don't have money to pay anyone, it's fine to ask for volunteers; just be clear that there's no compensation.

Some sprints have hired an experienced Sprint Master or professional editor.

Writing Resources

Make sure you have enough of the following:

- Extension power cords and boxes
 - Pens and paper
 - International power adapters
 - Chairs
 - A printer and printer paper
 - Ethernet cables or a wireless router
 - Mouse pads
 - Reference material
-
- A data projector (beamer)

Usually, participants bring their own laptops. Make sure to ask them to bring a laptop, or borrow one to provide for participants who don't bring one.

Venue

Venues are critical. You need somewhere that has enough space for everyone to work around one table. Discussions need to occur between writers so that you can negotiate the content. These discussions need to happen spontaneously and without having to find people in different rooms, etc.



The table needs good chairs, good power accessibility, and enough space for coffee cups, etc.



It's generally a good idea to make a rule against eating at the Book Sprint table. This is because noisy eaters can be very distracting to other writers.



Drinking is usually OK as long as no one is in danger of having his laptop showered by coffee.



The venue should be easily accessible to the local host and crew. Ideally, you want everyone to live in the same house that they are writing in. Living together can create bonds and friendships between people in a way that working together and leaving for the day does not. The aim of the sprint is not just to make good content but to make ongoing collaborations around that content--so get everyone in the same house if you can.



The venue should not be in the workspace of any of the main participants. It is important to make writing the priority, so the participants should not be disturbed by usual workday activities and distractions.

If possible, the workspace should be available 24 hours a day by all the participants. Someone may have a sudden late night inspiration or get up early because of jet lag and want to put in a couple hours of productive labor.

A fast Internet connection is necessary. The hub must be easy to connect to and must not block any content or protocols. Wireless is nice because people can move to other parts of the venue and still work, although you want them in the same room most of the time.

If you do intend for everyone to stay in the same house, sleeping quarters should be comfortable. Authors get the best beds, and organisers accept less comfortable accommodations. Bring extra ear plugs for those who may need to share rooms with chronic snorers. There should be a kitchen where people can get a snack if they need it.

Holding a book sprint in someone else's official space--a conference, a corporate office, a university, etc.--can work, but you sacrifice the simplicity and lack of distractions that you get by putting everyone in a house together. Official settings such as corporations can offer easy access to the Internet--but make sure nothing is blocked, because the participants need access to the outside world for references and communicating with others--and comfortable facilities such as a kitchen (but make sure you won't annoy regular staff by using it).

If you need to bring participants to some space like this in another facility, make sure to provide lots of signs. They need to easily find the workspace, kitchen, and bathroom. Ask the local host to go over the space in advance, plan where signs should go, and print signs with print as large as possible.

Lastly, if you can make the venue really interesting it can add a lot to the experience. Perhaps the most amazing venue we have worked with so far was De Waag in Amsterdam which is a kind of castle in the middle of Amsterdam. This was for many a once in a life time experience.



Tools

Tools for a book sprint consist of both hardware and software. It is best for each writer to bring or borrow a laptop computer. Software requirements are simply a web browser and possibly a screen capture or graphics software application. It is preferable, but not essential, that you work with Firefox when using the FLOSS Manuals web authoring tools. Internet accessibility is also required in order to access the FLOSS Manuals online writing tools.

Depending on the software or concepts you document at the Book Sprint, you may want to ensure that each writer can run the software or hardware. For example, at the OLPC and Sugar Labs book sprints, we borrowed enough XO laptops for each writer to have one for testing procedures and taking screenshots. In some cases we used USB storage devices to transfer the screenshots from the XO laptop to the writer's computer.

Make sure you have computers running the various different operating systems that run the software you are writing about. For example, if you were writing a manual about GIMP (image editing software) it would be good to have at least one Mac OS X machine and one Windows machine, in addition to Linux machines.

Collaborative authoring and publishing

The FLOSS Manuals site is a wiki, which enables simultaneous editing of content. (However, each web page can be edited by only one person at a time.) The wiki has a WYSIWYG interface with the option (usually not needed) to switch to an HTML view and edit HTML directly. The interface also lets authors quickly create and organize sections in a drag-and-drop manner using the Index tool. The site also offers a print layout that creates a print-ready (book formatted) PDF file. That PDF file is uploaded to Lulu for print-on-demand printing of the completed book.

Status indicators

FLOSS Manuals uses the following statuses to indicate to other writers the current state of the chapter.

- unpublished
- published
- hold
- complete
- incomplete
- needs images
- to be proofed
- needs layout
- to be translated
- needs updating

The published status is set for each chapter once a book that contains the chapter has been published by a maintainer.

Only maintainers can change the status of chapters, so make sure that you have one person with FLOSS Manuals Maintainer privileges at the sprint. One common approach is to grant everyone on site Maintainer privileges for the duration of the sprint (only FLOSS Manuals Administrators or Developers can grant Maintainer rights).

Permissions

Writers and editors must register for a FLOSS Manuals account before they can use the tool set.

Remote Contributions

In our experience, the majority of the work in a Book Sprint occurs in the real space of the sprint. There have been exceptions to this: for instance, the 'Introduction to the Command Line' sprint held in collaboration with the Free Software Foundation had few real space participants and many remote participants. Whichever way the participation is split, remote contributions are very important and it's absolutely necessary to devise a plan to co-ordinate the real space and online contributions. FLOSS Manuals provides some tools that have proven very useful in assisting with this communication.

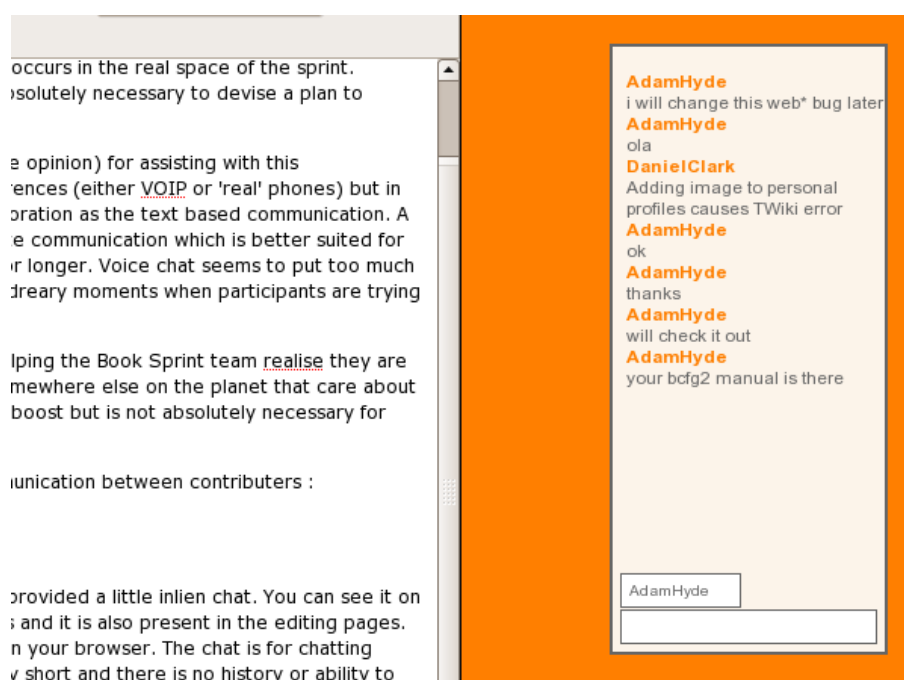
We have in the past also used phone conferences (either VoIP or "real" POTS phones) but in general, these are not as effective for ongoing collaboration as text-based communication. A persistent, open, text-based chat is good for "ambient" remote communication, which is better suited for collaboration with remote participants over a whole day, week, or longer. Voice connections put too much focus on "what are we doing right now" and there are often dreary moments when participants are trying to think of something to say.

However, quick conference calls can be very effective in helping the Book Sprint team realise they are not working in isolation and that there are active, real people somewhere else on the planet that care about what they are doing and want to help. It gives a motivating boost but is not absolutely necessary.

FLOSS Manuals provides the following tools for online communication between contributors.

Inline Chat

To exchange brief messages with others while contributing to a manual, we have provided a little inline chat. You can see it on the right of all pages in the WRITE pages of the FLOSS Manuals as well as when editing pages. It is a simple mechanism requiring no extra plugins to work in your browser. The chat actually works by sending messages to the FLOSS Manuals IRC (Internet Relay Chat) channel (see below). The chat is for chatting about anything that comes to mind. The text format is pretty short and there is no history or ability to scroll back. So for more elaborate conversations where you want to record the discussion or allow private messaging, an IRC client should be used.



The inline chat is a single chat room. There is no provision (yet) for chatting on a manual-specific basis. All comments are out there in the open for all to see and remain there until someone else starts a discussion and

makes your comments scroll off the page.

IRC

IRC is a very old school technology. Its main inhabitants are geeks, especially free software geeks. It's a very good technology for chat as it requires very little bandwidth, so if you are on a slow connection it works very well. IRC chat rooms are provided by IRC networks, servers that offer accounts. We use the Freenode IRC network, one of the oldest and most popular around. Anyone can create a chat room on the Free Node network. we have created two:

1. #flossmanuals
2. #booksprint

The web-based chat described in the previous section connects to the first channel (#flossmanuals). We hardly ever use the #booksprint channel.

To use the IRC chat rooms without using our web-based interface, you need to have an IRC client installed and know how to use it. Connect to #flossmanuals at *irc.freenode.net*.

Email Notifications

Every manual has an email notification feature. By subscribing to the notifications, you will get an email every time a change is made in that manual. These notifications also have a link to a "diff" view, which will show you the changes made in the most recent edit and highlight the differences.

Email notifications are great during book sprints for keeping an eye on who is doing what. In addition, if you have a problem or question with a change someone makes to something you have written, you can find out about the change quickly and contact the person to discuss the reason.

Editing Notifications

In the editing interface you can see who is editing what chapter.

Chapter List		Actual File Name	Edit
3% complete			
FIREFOX			
INTRODUCTION			
INTRODUCTION TO FIREFOX	edit	needs updating	Introduction
INTRODUCTION TO OPEN SOURCE	edit	unpublished	IntroducingOpenSource
INSTALLING			
WINDOWS	edit	needs updating	InstallingWindows
OSX	edit	needs updating	InstallingOSX
UBUNTU	edit	needs updating	InstallingUbun
UPDATING FIREFOX	edit	incomplete	UpdatingFirefox
UNINSTALLING FIREFOX	edit	incomplete	UninstallingFirefox
INTERFACE REFERENCE			
INTERFACE HIGHLIGHTS	edit	needs updating	Interface
LOCATION BAR	edit	needs updating	OpenAURL
SEARCH BAR	edit	unpublished	SearchBar
ACCESSIBILITY FEATURES	edit	unpublished	AccessibilityFeatures
BROWSING BASICS			
NAVIGATING WEB PAGES	edit	to be proofed	NavigatingWebPages
COPYING SAVING AND PRINTING WEB PAGES	edit	to be proofed	CopyingSavingPrintingPages
SEARCHING	edit	to be proofed	Searching
BOOKMARKING SITES	edit	unpublished	BookmarkingSites
LIVE BOOKMARKS	edit	needs updating	RSSLiveBookmarks
BOOKMARKING FILES	edit	unpublished	BookmarkingFiles

This is useful for seeing how much activity is going, and helps co-ordinate contributions. If someone is editing a chapter, no one else should edit that chapter until its free. Otherwise edit conflicts can occur.

Finally, when you are in the WRITE section of FLOSS Manuals, you can see everyone who is currently logged on. These people should be able to see anything posted in the web-based chat.

Tips for sprint writers

Encouraging writers who don't know they are writers yet

With a Book Sprint, often you will encourage people to write who are very enthusiastic about the project but don't have confidence in their writing skills. The general attitude of a Book Sprint comes from the energy of the writer participants, so it is important to foster a positive attitude that there's a lower barrier to entry and that all contributions are welcome, no matter how small in size or uncertain in skills or experience. The participants will bring various experiences with them to the Sprint. Some participants will be great testers, editors, or plain readers who offer comments and suggestions. Professional writers know certain tips and tricks of the trade, starting with a style guide.

Make sure you create a link to a style guide, or agreed-upon standards and terminology is helpful for writers to know why edits are taking place. Also, writers who come and go can be confident that their edits and new chapters match others. A style guide also helps remote contributors who can look up a question before they ask it.

Since wikis tend to have one "winning" page - the current one - be sure you have an arbitration policy in mind before you start. Is it benevolent dictator, a single editor, or an editorial board?

I must emphasize the importance of planning ahead and getting participants or the project principles to determine and agree to an audience. Coming to agreement on this basic question "Who will read this text?" can save you many discussions later. The audience selection helps you determine focus for the writing because different audiences have different tasks to achieve.

If possible, fill out the outline ahead of time and base it on task analysis for the selected audience. This method is called "scaffolding" because it offers a framework within which the writers can construct the book that is the outcome.

Try to train writers on the tools ahead of time, by offering a training course or screencast. Even though this book focuses on FLOSS Manuals tools, other collaborative authoring tools may work well for a book sprint.

Adams Thoughts

Here's some thoughts from my experience leading sprints. Much of my inspiration for running events comes from seeing Allen Gunn (Gunner) in action with Aspiration Tech events. My experience in running these events has been a process of taking the intent of Gunners methodology and working out a way for it to work within a different context, namely the collaborative authoring of comprehensive text. Although I haven't tried this yet, I am very sure this method has broad application beyond the writing of Free Software manuals.

Writers vs Contributors

I prefer to refer to people that contribute to a manual as a 'contributor'. Sounds pretty straight forward but it can be tempting to use the term 'writer'. This, I think, is a mistake as the central tenet of FLOSS Manuals is that anyone can contribute to the Free Software movement, and they can do this by contributing to free documentation. Contributing can mean adding images, writing, editing, checking for technical accuracy, copy editing, hosting sprints, organising sprints, designing icons, designing covers, spell checking, changing the tone of a sentence, or reading the documentation and asking questions.

When describing the task at hand at the beginning of the sprint I prefer to use the more inclusive term 'contributor' than 'writer'.

I also prefer not to use the term 'author'. Although I respect the tradition, the term has taken on unnecessary baggage. Except for the few enlightened authors the term implies someone who works for a publisher, participates in a publishers star system, owns the copyright for their texts, licenses content as 'all rights reserved', and chases away attempts to alter their work. There is no place for 'authorial' attitudes of this kind in a collaborative writing environment. More enlightened authors, or those wanting to broaden their horizons are always welcome!

Team Development

One of the most interesting things about Book Sprints is the intensive social experience. This not only leads to very good conversations, which in turn lead to good content, but it is very common for strong bonds to form amongst the team.

Books?

I think a more accurate term is 'comprehensive text' - this is really what FLOSS Manuals is trying to produce. One storage and display medium for this material is a book, another is a webpage etc etc. Still 'comprehensive text' is too unwieldy a term so I don't use it much.

Publisher?

What is a publisher these days? I have no concrete answer to this. All I can say is that FLOSS Manuals is not a publisher in the traditional sense, we are working across many paradigms, and mixing them up as needed. That leaves us somewhere inside the publishing sector but outside the traditional processes.

Living Texts

In my opinion a text has a better chance of being born and living if it is collaboratively written. Often authors feel protective about their text and this often hinders the ongoing development of the work. Hence this kind of material is more likely to 'die' from the author protecting it whereas a text that has been sourced from a collaboration has a better chance of living because there is no one fending off contributions.

So when you start a sprint make sure you emphasise that all authorial egos must be left at the door.

To this end it is necessary the any text generated is created under an open license.

Food and Venue

If you provide a good venue, and good food you will not only have a bunch of happy sprinters, but they will want to do it again :)



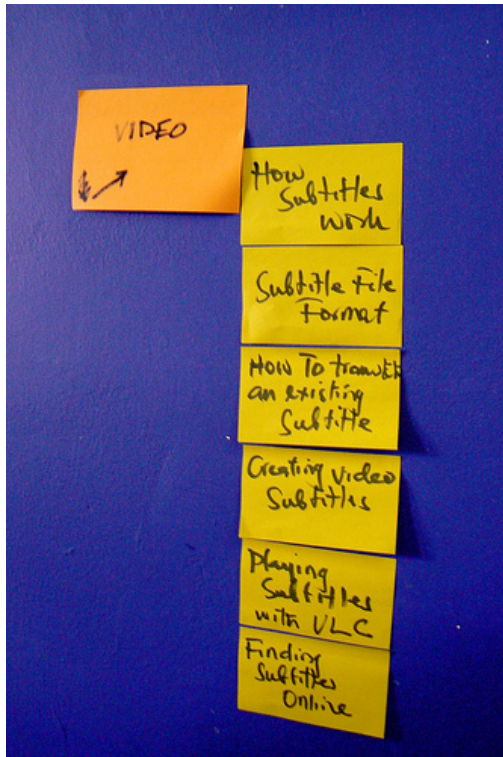
Creating the Index

The Index is the skeleton from which a sprint must build a body of content. Unless it is impossible, the Index should be created by the people at (in the real space of) the sprint.

I have found that when we meet on the Sunday night before the sprint we can top the index off in about 3 hours. Having everyone at the sprint involved in the process of creating the Index is important for the following reasons :

- everyone at the sprint understands the job at hand
- it creates buy-in by all participants
- it creates a good dialog from day 0

The Index should be necessarily vague. It should just be a series of section headings and chapter headings. You can use the Index tool in FLOSS Manuals to create this or you can use sticky notes and put them on a wall with the section headings and chapter titles.



Anything with more detail will only confuse the team. The content should be worked out dynamically as the sprint goes. The sprinters should just be able to write on Monday, 9am with a clear head. It's far far easier to get them started and bring everyone inline as the week goes, rather than shackle everyone with constraints on what must be written where and when - this is my strong opinion, based on the experience of many sprints, but i do acknowledge that I haven't tried the later approach. My feeling is that it might work in some instances but you would have to think about this carefully.

Combining another Event

For the Open Translation Tools Sprint we followed a 3 day conference (hosted by Allen Gunn and Aspiration) with a 5 day sprint. Since the topic of the conference and the sprint were the same we decided to integrate the two as much as possible. For this I built an outline on the wall of the Index with sticky notes and asked people to add/subtract/change the outline as they saw fit. At each session I also demonstrated the FLOSS Manuals tools so that those that did not attend the sprint could get enthused and participate remotely.



The methods we started to develop here can be taken a long way but I learned a few things from this experiment. This first is that about 10% of those that commit to participate remotely actually do - which is still not bad. The second is that many people contributed substantial material that they had already written - this was great! Although we did not use much of this material some was very useful and very well written. It took 2 days of non-stop reminding people of what we were doing for some of this content to emerge so don't give up!

Narrative Flow

The narrative flow of the text should be simple. Keep the text linear, progressive and broken down into clear standalone chunks. Each chapter should follow on from the last and lead to the next. Fundamentals at the beginning, leave no term unexplained, and ramp up the complexity as the book progresses.

If you do this the newbies can start at the beginning and the experts can jump into the book from a later starting point.

Do not make the mistake of over complicating your narrative. For example, the idea of weaving thematics or case studies through multiple chapters is going to make it harder to make the text harmonised and it will likely slow down the generation of content. Also, any interweaving of themes that requires dependencies (ie. the requirement to read specific chapters before reading a chapter) will not be able to be re-used within another context very easily.

Getting People Started

Enabling written contributions from people who do not see themselves as 'writers' is a very important part of the Sprint process. There are a few important ingredients, the first is that people should feel comfortable, unstressed, and in a good mood. Its not always possible to manufacture this, but making it true for as many of the participants as possible is a necessary role of the organiser. Secondly, people should write about what they know. Like talking, writing is easier if you know intimately what you are talking about. Also, like most things, its easier to keep people going if they are allowed to get immersed in their subject.

Inorder to do this, it is very important that participants start the sprint by choosing the chapter that most interests them. Let everyone start off in a somewhat random manner - don't worry about this, over the course of time the writing will become more co-ordinated. It is something like the start of a formula 1 motorbike race - all the riders run for their bikes and just get going, it seems like chaos at first, but over a short time things sort themselves out. The important thing is that people just get into it, forget their personal belief that they can't write, and just start writing.

For this reason also, it is important not to overburden the beginning of the sprint with writing rules. The process of improving writing style should occur organically - this can happen through discussion with the other participants and each individual contributor thinking about their own style, looking at the work of other, and changing over time, or the writing will improve by others re-writing content or the editing process.

So, start as unburdened by rules as possible. Of course, someone will likely ask 'where is the style guide' - for this purpose each manual has a style guide but it is linked to another page and not a very present part of the manual home page. So, those that want rules, can have them, but everyone else that doesn't want to think about these issues can safely ignore them.

Writing as Conversation

"Writing as a conversation" is a statement by Anne Gentle. I can't speak for Anne, but for me this statement has a very clear and important value in Book Sprints. The conversation takes place on at least four levels :

1. **the conversation in the room.** With all participants seated around the same table conversations need to be fostered. Unlike traditional writing processes, this is a noisy environment and chatter should be encouraged. It is through these across-the-table conversations that the participants are harmonising their ideas about what they are writing.
2. **the conversation in the text.** The texts themselves are conversations. Do not be protective about the content you contribute - encourage others to read it, change it, delete it etc. The continual combing through of the content by many eyes ("Many eyes make bugs shallow") is a fundamental tenet of Free Software and serves equally well in this environment. This constant changing is itself a conversation happening in the text and it is essential for improving and harmonising the content.
3. **the conversation outside the room.** Be in touch with people outside the room and ask for their take on the texts. It is important to know if the texts really are going in a direction that can be understood by people that have not been privy to the conversations in the Book Sprint room. Encourage remote participants to feed back into the conversation via IRC, email, editing, and the FLOSS Manuals web chat.
4. **the conversation with the reader.** When writing, use a direct and friendly and conversational voice. When I worked in radio one trick to getting DJs to relax and sound 'normal' on air was to cut out a picture of a face, stick it to the microphone and ask the DJ to talk 'to' that person. This worked. We tried a similar approach for the Circumvention Book Sprint by building readership profiles but I don't think it was effective (but it is an intriguing idea and I would like to experiment more with this approach). Andy Orams approach of writing a sample chapter with the desired tone is the best experiment yet for us in this area but we need to test it more. The same effect can possibly be achieved by encouraging contributors to think of the reader as someone they know well and respect.

Disruptive Conversation

Occasionally you might need to deal with a disruptive presence. My experience has been that there are people who are so passionate about the topic at hand all they want to do is talk about it, and talk about it, and talk about it. This is not actually helpful. Although writing is conversational in this environment it is not a soliloquy or a monologue. This kind of behaviour can be very disruptive and you will need to talk directly to the person concerned. This might be in open if done directly, but more effective is a casual but direct and clear communication with no one else around. Most likely if you break this issue gently the person concerned will listen and make good.

Gaining Trust

Its very likely that the group assembled to write a book has not done a Book Sprint before, or they may have never participated in a process orientated event. There is a good chance then that some of the participants will feel wary of this seemingly unstructured methodology. When you deliver the results this attitude will change - but that is at the end of the week...how to get through those first few days? Well, the only thing that I have found that will do it, is to feel strong within myself that I help will get them to the finish line. So, you have to feel confident. Don't let any nay-sayers get to you. If you don't have any tricks to help you through this then find them fast as you must feel confident when kicking it all off.

As for the nay-sayers. My preferred strategy is to joke with people to loosen them up. If someone is being uptight, its a good idea to gently make them conscious of this - not in a direct way, but with a friendly joking. The point is to chill them out, not to sit down and have a serious conversation about their concerns.

The Shape of the Week

So far my experience has mostly been with 5 day sprints with everyone arriving on Sunday and leaving on the following Saturday. Ideally we have a drink on sunday night and finish the index development. Monday morning is a quick tools demo and then we get into writing immediately. We write until Wednesday night, and then after Wednesday we freeze all chapters - no additional chapters can be added unless there is a

absolute necessity. This means Wednesday night is usually the hard talk about what will stay in the manual and what will be thrown out. Everyone has to commit to finishing particular chapters by the end of **Thursday**. The Wednesday night talk also will include a slight restructuring of the manual to fit in line with any inconsistencies discovered and not resolved during the week.

Thursday is sprint within a sprint day. Everyone should write until they can't write no more. The aim is to get all content finished by Thursday night.

Friday is clean up day. 6pm open some drinks, make the pdf, upload it, buy a book, drink the drink.

Pairing it up

A great strategy that Michael Mandiberg (Digital Foundations) introduced into our process is 'pairing up' experts and non-experts.



Having an expert talking through concepts while the 'non-expert' asks questions and writes things down in a 'human readable' format can be very effective. This gives the newbie something to do which is more involved in the content than spell checking, and they learn a lot at the same time. It is also great for an expert to be confronted with difficult newbie questions - it forces them to confront their audience directly and break concepts down to clearer explanations.

Structural jams

You will surely get into a jam with the structure of a chapter, or the structure of the entire manual during a sprint. I recommend dealing with these issues as soon as possible. I have several review processes over the first few days - usually one at the end of the first day, and one at lunch the following day. The issues can be left to these meetings. Later in the process (after the second day) deal with issues immediately. If someone brings up an issue about the structure get those involved and any extra heads available to work it out.



A good strategy for reworking the structure of a chapter is to print it out, get a marker and some scissors and cut it up ...not really ala William S Burroughs, but in a more linear fashion. Chop out bits, rearrange them, cross out material that should not be there and then reflect those changes in the manual. For this process it is good to have someone take the lead (it does not necessarily have to be the sprint leader) and bring in people who are and are not familiar with the content of that chapter.



Harmonising Flow

Generally speaking the flow is taken care of by the conversations that occur in the room and the cross-editing that evolves from this. However in the event that you are working with a larger group the content generated might be too broad for much cross-editing and discussion.

To get a good flow between chapters in this situation I think its a good idea to assign the editing of a section to one specific person. After all the content has been written for a section get someone to read all the chapters and restructure/edit to make it sound like one person has written it as an evolving narrative. They may wish to write introductions and endings to each chapter to help connect the flow. Be careful however not to make these connections to specific as the material has to make sense if it was read by itself or in another context.

There is no problem in then getting another person to go over the same material with the same brief. The more people you have going through 'complete' content, the better it will get.

Tone

I think manuals should be a good read. Generally speaking the tone should be chatty and engaging. I know many technical writers would prefer more procedural texts but I think this kind of information is better left for quick help wikis etc. FLOSS Manuals creates comprehensive texts and this is better suited for chatty discourse than stripped down help systems. Here are some examples of tone that I like in manuals :

Example 1

The first paragraph of the opening chapter of the FLOSS Manuals Introduction to the Command Line manual starts like this, note the very direct but intimate tone in all of these examples.

```
"Commands enable you to do all sorts of powerful things on your
computer. We'll demonstrate this by looking at an everyday task that
might be familiar to you. If you use a digital camera, you probably
have a folder full of images on your computer. Imagine you wish to
resize the image profile.jpg so it is 300 pixels wide and save it as a
new image called profile_small.jpg.
```

Using an image editing software, the steps you need to go through might look as follows:"

Example 2

From the How to Bypass Internet Censorship manual; an explanation of who controls the net :

```
"The full story of Internet governance is complicated, political and
still being actively disputed. This text is meant to provide enough
details to help you understand how certain aspects of the system affect
particular methods of restricting access. The key point is that, in
some countries, all Internet infrastructure is owned and operated by
governments and large regulated telephone companies. A government that
wants to block access to information can exercise direct or indirect
control over points where that information is produced, or where it
enters or exits the country. Governments have extensive legal authority
to spy on citizens, and many also go behind what the law allows, using extra-legal methods to mon
```

Serendipity

Book Sprints are more art than science. Don't make the mistake of over-formulating them. It is absolutely necessary to leave some chance in the system at every step except where organisation logistics are required (venue, travel etc). For the rest, who attends, the scope of the book, how the book will be written etc - cannot be strictly formulated anyway so learn to coach a dynamic process on the fly and resist any indoctrinated impulses to micro manage. If you do micromanage the quantity and quality of material produced will suffer.

Also, it is amazing what can happen if you let it.

Follow Up

I recommend you consider getting the book out there asap while the energy and enthusiasm within the sprint team is at its peak. My recommendation would be to do a quick edit, don't think too hard about it, check some of the formatting issues mentioned below, and get it out there. If you can commit to really pushing it by the end of the week you will harness the energy you have all built up during the past weeks. The longer you leave it, the greater the amount of energy that will dissipate and the less effective your marketing for the book will be. Things that need to be considered immediately after the sprint include :

1. **Who will be the maintainer(s)?**

Its a good idea to appoint someone (or more than one person) to maintain the manual - this doesnt mean writing the manual, but drawing others into the process and keeping an eye on updates. At appropriate moments this person should update the sources on both the static HTML manual, and the book sources on lulu.com

2. **Promotion**

The books uptake, and the value that the user base gets from it, is directly proportional to the profile and push you give it. My recommendation is that you use all channels that are personally available to each of the sprinters. My recommendation is to make it as high profile as possible. We have noticed significant improvements in uptakes of manuals when they are linked directly from the home page of the project. Personally, I feel if you can really push this the additional 'look what we did last week!' angle, you have a nice story to catch peoples imaginations. The longer you leave it, the less effective this story becomes. Fast results are also good for acquittal purposes if you have to show results to funders.

3. **Bookstore widget**

FLOSS Manuals has a bookstore widget (see <http://www.flossmanuals.net/bookstore>). You can put this in your website to promo the book. It provides a link to buy the book immediately (or download the free PDF).

4. **Formatting**

The printed manual will probably need a little formatting. If anyone wants to get stuck into this they can output book formatted PDF to check via Objavi:
<http://www.flossmanuals.net/bin/view/objavi>

Inkscape Book Sprint

The first 'full featured' FLOSS Manuals Book Sprint was for the Inkscape manual. Inkscape had very little documentation which was surprising for such a mature project. We gathered a team of people together from the USA, France, Russia, Australia, and Amsterdam - most of these already comprised the 'official' Inkscape documentation team. We were 'shooting in the dark' with this one as the process was entirely unknown...Book Sprints were at this stage, just a mystery.

Since this sprint we have tried to add something new to the FLOSS Manuals interface to assist the sprint process, but for this sprint we just used the basic version of FLOSS Manuals as it was then.

Funding

Thankfully the costs were covered by Google from their Summer of Code project (big thanks to Google and Leslie Hawthorn).

Invite List

The process was fairly simple. FLOSS Manuals in co-ordination with Joshua Facemyer (Inkscape Documentation Team) invited those active in Inkscape documentation to attend via personal emails and an email or two to the Inkscape Documentation email list. Everyone said yes!

Location

We held the event in France once it became clear that most of the participants were from France. The event happened in the Cite de Sciences in Paris. Although the sprint was extremely productive it was our first one and we could have avoided some simple mistakes had we understood the process a little more. I think we suffered a little from the accommodation being over an hour away from the venue and those attending from out of town were staying in four different places. Also we had partial access to the space since security would check and kick everyone out if it was past 10pm. Additionally the access to food was limited.

However, this was our first sprint and Elisa did a fantastic job of organising everything - its just that we didn't know what an optimum sprint required at the time. Despite this, it was a very successful event which goes to show that despite our inexperience, we still produced a lot of excellent material.

Much of the content actually already existed in French and was translated during the sprint to English. For this sometimes two people would sit together - a native English speaker, and a native French speaker - and work through the translations together.



Lasting Outcomes

The sprint team still works together remotely and a very strong French language team continues to collaborate on the translation of the manual. Joshua Facemyer (pictured left) also attended the Digital Foundations Book Sprint as the Inkscape and Scribus expert. Also, during this event Brianna Laughher (pictured right) produced an excellent manual on how to contribute to Wikimedia Commons.



XO Sprint

This BookSprint grew from an idea that a manual for the next Give One Get One release from the One Laptop per Child project may help with supporting people who purchase the laptop as if it were any other consumer product. In reality, the goals of the the One Laptop per Child project are to educate children, not to build consumer products.

The OLPC project aims to produce a low-cost, low-power laptop for students in developing areas of the world. Translation to additional languages is an important aspect of the project, and participants should learn to embrace open source philosophies of allowing use, changing, and modifying of the source, including instructions, information, and knowledge, for the software that runs on the laptop.

The Book Sprint became a workshop where subject matter experts shared their knowledge of the products, and writers shared how to write good user documentation. It was also a social experience where attendees formed a community of shared common goals and experiences. For the OLPC Book Sprint the attendees gave (up to) a weekâs time to be curators of information housed in wikis and websites everywhere.



Funding

With attendees coming in from around the world and the need for them to sleep and eat while they write, it was necessary to finance their accommodations and travel. With three matching \$1000 support pledges from vested organizations (RedHat, OLPC, and an individual donation to SugarLabs) plus the fundraising efforts of FLOSS Manuals, we provided travel and lodging and some meals for contributors. With five or six in-person contributors and at least twice that many contributing remotely, we were able to fund the BookSprint for about \$4000 total. There are two books for sale as a result of the BookSprint, and a two-Euro markup on each book should help pay for these type of maintenance costs going forward.

Platform Development

For this sprint Aleksandar Erkalovic (aco) added status notifications to the editing interface. This meant that we could mark a chapter with an individual status such as 'to be proofed', 'needs images' etc.

The status markers worked well but provided a heavy load on the server due to the way we implemented them. Aco, as always, came through and rewrote the mechanism to make the load much lighter. The status markers were wonderful and assisted the process a great deal. Now contributors, whether remote or in real space, could see what needed to be done for each chapter at a glance.

Invite list

Attendee selection and invitation or allowing for organic invitations is an important first decision. After you determine your core set of participants, you choose a date and location that should work well for many if not most of them. In the case of our FLOSS Manuals BookSprint for OLPC, a flurry of activity on several mailing lists initiated the desire for a BookSprint. Invitations stemmed from those who were initially interested but also from community connections already made among OLPC enthusiasts.

Location and date selection

Location and date selection go hand in hand because some locations are not going to be available for certain dates. Travel time, distance, and cost as well as visa considerations are part of balancing this equation as well. When someone needs a visa to come to the U.S. to participate in such an event, at least a month's lead time is necessary.

Some host locations will want all participants to sign a release so that they are not liable for any unfortunate result at an unconference or BarCamp. Signing a release should be fine — nearly all this planning is happening on the good faith in others anyway, so suing when the location is typically given for free would be poor form.



Outputs

We made 7 manuals during this sprint.



Two of these became books of 120 pages (about the laptop) and 180 pages (about the operating system - sugar) each.



Since then we have remixed content into a third 'giant' 280 page book. This is the advantage of the FLOSS Manuals system - being able to recombine material in any number of ways to create new outputs. A remix was also created for distribution on the laptop itself for the roll out in December 2008 (which is ongoing) and the OLPC crew build a helper application to display the manual from within the laptop, so now all laptops come with the documentation created at the sprint. The documentation on the laptop has been taken directly from the FLOSS Manuals remix and downloaded as a zip file - a process taking about 5 minutes. This zip was then included in the master version of Sugar.

about style guides for open source projects is that the fewer rules writers have to remember, the better. However, this parsimony must balance against readers' need for consistency.

After that, I reviewed chapters in the XO and Sugar manuals that were finished or nearly so, making sure they complied with the style guide, and editing anything I thought needed it. I tested procedures with one of the XO laptops that were scattered around the room. Apparently, this made an impression on David Farning, a Sugar programmer who attended the BookSprint and who later wrote: I realized this was not just a couple of programmers trying to throw together a wiki as I watched Janet Swisher intensely studying the XO's battery. Turns out she was trying to determine if the "installing the battery" section could be misread. From my experience, a programmer would have said, "If they can't figure out how to put the battery in, what's the point of a fine manual?"

One factor that I did not have to worry about for the BookSprint was writing tools. The FLOSS Manuals Website, being a wiki with a WYSIWYG HTML editor, was dead simple to use. The site even kept track of which chapters were currently being edited by whom, so that we wouldn't step on each other's changes. There were times when a chapter that I wanted to edit was one of several that the system said was being edited by another person. Since we were sitting in the same room, I could just say "Hey, are you still working on that chapter? If not, can I break your lock?" For certain sections, I decided that the most appropriate HTML structure was a definition list, which was not supported by the WYSIWYG editor. Being familiar with HTML, I was able to switch into "code" view and insert the tags manually. However, within a few minutes of my mentioning this to Adam Hyde, he had modified the editor, and buttons for the needed tags appeared on the editor's toolbar.

Another benefit of co-location was that when I came across a note that said, "Walter, check for accuracy," I could just say "Hey, Walter," and Walter Bender (founder of SugarLabs, former president of OLPC for Software and Content, and former executive director of the MIT Media Lab) would walk around the table to look over my shoulder. Talk about immediate feedback!

Bringing together communities with related interests can lead to unexpected synergies. The Sugar API documentation was not within the scope of the BookSprint, but when it came up in conversation, I was able to point the Sugar folks toward a tool for wikifying it, to encourage programmers to contribute. Also, at the Wednesday cookout, my husband, who is a musician and a graduate student, got into a long discussion about computer-generated music with Adam Hyde, who is a "sound artist" as well as the FLOSS Manuals impresario. (OK, that has nothing directly to do with the BookSprint, but it wouldn't have happened without it!) In comparing this and other face-to-face meetings of virtual communities, I can see the following benefits:

- Community members get to know each other "for real." If you've ever worked on a virtual team, you know that you work much better with team members you've met face-to-face. Putting faces to names and personalities to email addresses helps the team or community work together virtually in the future.
- A concrete time and place to work increases productivity tremendously. Volunteers working asynchronously tend to lack urgency; there is always some other priority that pushes the volunteer work off to "someday." At a sprint, the work must be done here and now. And, as I mentioned, feedback can be provided within a few seconds instead of hours or days.
- The event helps put the project in perspective for participants. When a project is coordinated online, participants can get swamped by details. A sprint helps focus priorities and helps participants see how their pieces fit into the bigger puzzle.
- It's fun! People feed off each other's energy and excitement. Jokes are cracked, camaraderie develops, and friendships form.

FLOSSify

"FLOSSify" is the name we give to Book Sprints where the purpose is to convert an existing book into a book about Free Software. Our first FLOSSify event was with the design text book "Digital Foundations". This book was published in December 2008 and the books authors, Michael Mandiberg, and xtine burrough had negotiated a Creative Commons license with the publishers (Pearson) - this made it ripe for a FLOSSification.



The original content was about teaching Adobe CSS4 by teaching students Bauhaus design principles. Michael and Adam Hyde worked closely together to organise an event in February 2009 at Eyebeam in New York city. The event was 3 days long and was an open door for anyone wanting to walk in and assist. We announced it on the FM blog, and through Eyebeam channels and other personal channels.

In assistance was also Patrick Davison.

Platform Development

Aco developed a very nice new tool for this sprint that illustrated who was editing which chapters in real time. In this was we would see orange flashing boxes next to each chapter name, and the boxes contained the name of the person editing the chapter at this time. This avoided conflicts, but more importantly the system made you feel like there was a lot happening as you could immediately see who was actively working on the manual with you whether they were on the same room or working remote.

TYPE ON THE GRID	edit	needs layout	TypeOnTheGrid	
COLOR THEORY	edit	needs layout	ColorTheoryAndBasicShapes	
LINE ART & FLAT GRAPHICS	edit	unpublished	LineArtAndFlatGraphics	MelChua
FIREFOX				
SEARCHING AND SAMPLING	edit	unpublished	Searching	
GIMP				
SCANNING	edit	needs layout	Scanning	
TONAL SCALE	edit	needs images	TonalScale	ChristopherBlount
LAYERING AND COLLAGE	edit	unpublished	LayeringAndCollage	DaveMandl
REPETITION AND CLONING	edit	unpublished	RepetitionAndCloning	JenniferDopazo
NON-DESTRUCTIVE EDITING	edit	unpublished	NonDestructiveEditing	
GRAPHICS FOR THE WEB	edit	unpublished	DigitalOutputForTheWeb	
SCRIBUS				
MULTIPLES: CREATING OF UNITY	edit	needs layout	MultiplePagesUnity	
MULTIPLES: CREATING TENSION	edit	unpublished	MultiplePagesChaosTensionDisarray	PatrickDavison
KOMPOZER				
HELLO WORLD	edit	to be proofed	HelloWorld	
FILES AND SERVERS	edit	to be proofed	FilesAndServers	
SEPARATING FORM AND CONTENT	edit	unpublished	FormAndContentOnTheWeb	AdamHyde
PROCESSING				

We also developed the simple chat so that discussion could happen about content in real time and from the FLOSS manuals interface.

MelChua
http://wiki.inkscape.org/wiki/index.php/Inkscape_for_Adobe_Illustrator

MelChua
in case it's helpful to anyone
(I'm an Inkscape native)

ChristopherBlount
tonal scale is done

ChristopherBlount
just needs images for camera
connection in linux

AdamHyde
nice work!

MelChua
line is done, except for img
uploads

PatrickDavison
hot

AdamHyde

The Sprint

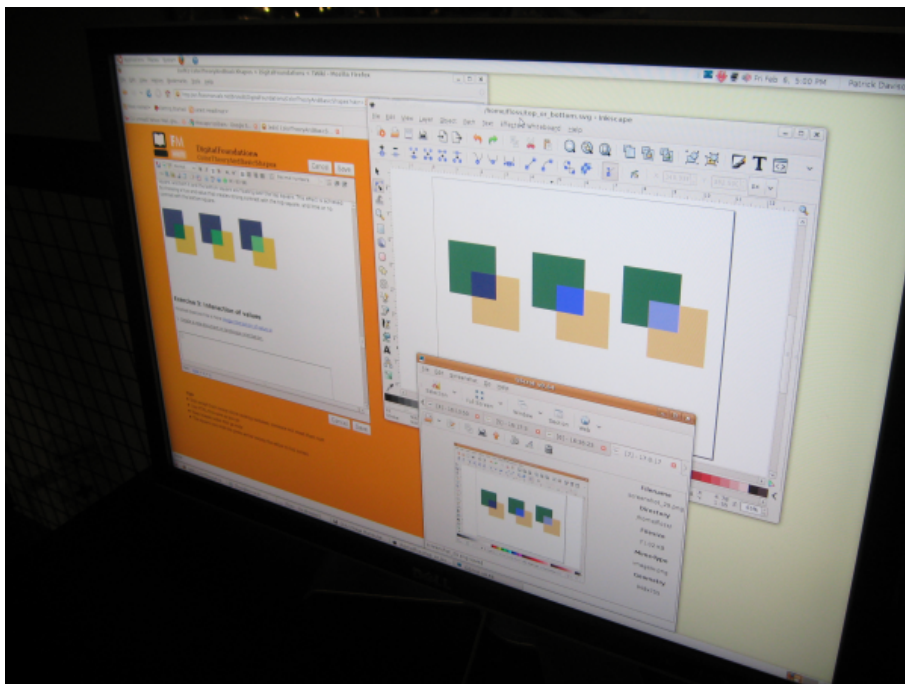
When the sprint was about to start we aligned ourselves with the task and made sure we had the same idea. Essentially the plan was to make the sprint fun, provide food, and maybe some drinks later at the end of each day. FLOSS Manuals also paid for a flight for Joshua Facemayer who was part of the Inkscape Book Sprint team in Paris.

On the first day we had 7-9 people in the room at Eyebeam. Many of the people that attended were unknown to us and some, for example, Chris Blount made huge contributions.



The process of converting a text book from non-free to free software might sound easy...just replace the examples right? Well right, except that the paradigms for the softwares are never the same, so the process is one of re-creating every example and rewriting all the content referring to the excercises.

So, for example, when rewriting a chapter on Illustrator, and replacing it with Inkscape, Chris Blount had to redo all the screen shots, work out each step, document it, and then rewrite the texts and replace the images.



To do this you need to know the software (Inkscape) pretty well...

We also worked with xtine remotely as she was based in LA, and the interface for remote collaboration worked extremely well for holding the team together.

In all it was an extremely productive and rewarding process and the results speak for themselves :

<http://en.flossmanuals.net/DigitalFoundations>

We are currently considering printing this manual/text book on newsprint and distributing throughout schools and universities.

Command Line Manual

"I have written basic introductions to the command line in three different technical books on GNU/Linux and read dozens of others. FLOSS Manual's "Introduction to the Command Line" is at least as clear, complete, and accurate as any I've read or written. But while there are countless correct reference works on the subject, FLOSS's book speaks to an audience of absolute beginners more effectively, and is ultimately more useful, than any other I have seen."

--Benjamin Mako Hil

Some wonderful feedback about this manual...and what is more this manual was written in 2 days.

This sprint grew out of discussions that Adam had been holding for a while with leaders of the Free Software Foundation was considered from the beginning to be just the first project in an ongoing collaboration. The FSF expressed strong satisfaction with the outcome, but so far no second project has been planned.

The unusual aspects of this sprint included:

- It was squeezed into a weekend. (One other sprint has been done in two days since then.)
- There was very little time before the sprint, so pre-planning was compressed.
- The sprint was held concurrently with an FSF annual conference, and many people participated while taking breaks from the conference.
- An usually large number of authors participated, and more than half were remote. (We have not tried to determine what percentage of the text was contributed by remote participants, but certainly it covers several complete chapters.)
- The topic was very familiar and had been covered by a lot of existing documentation (including a very thorough man page and info page about Bash) as well as by many commercial books.
- The outline was written by a professional editor (Andy) who happened to know the subject well also.

Platform Development

For this sprint Aco replaced the hacky chat Adam had made with a more functional chat which interfaces with the FLOSS manuals IRC channel. This means that anyone can chat in the same IRC room using either the web chat in the FLOSS Manuals interface or their favourite IRC client software. Utilising the IRC like this was especially important for this sprint since this is the preferred channel of communications for experienced users (ie. FSF super users).

We also added a mechanism for listing who is logged in at any given time.

Writers currently logged in :
AdamHyde, AlejoDuque,
AnneGentle, ClaraJo,
ElisaYemanja, JamesVasile,
JenniferDopazo,
JimmyVolatile,
JoshuaFacemyer, JoshuaGay,
MelChua, PatrickDavison,
SimonJolly, SkierPage,
WilliamLai, XtineBurrough

18 writers today:
AdamHyde, SkierPage,
MelChua, JenniferDopazo,
ChristopherBlount,
PatrickDavison, XtineBurrough,
JamesVasile, JoshuaGay,
JoshuaFacemyer, ClaraJo,
SimonJolly, WilliamLai,
AlejoDuque, JimmyVolatile,
ElisaYemanja, AnneGentle,
MichaelMandiberg

Pre-planning

When we planned the sprint in March 2009, we had decided that it was important to provide detailed outlines for sprints. We have subsequently revised that dictum, but in this case it proved quite valuable because there were so many people converging on the book for such a short time. Also, the outline was easy to write: Andy produced the first draft in about an hour, and although other people added a few sections, but the first effort was pretty stable throughout the whole sprint. A few more sections were added during the sprint, and after the sprint Adam rearranged the order of many sections.

As news came in that people would be helping remotely, Adam realized that we'd have an unusually high number of concurrent authors, and suggested that we break up chapters into small files. This was good advice. The result is that this manual is broken into an unusually large number of separate web pages: about forty web pages for a book of 130 pages.

Logistics (venue, food, publicity) were handled mostly by FSF staff. Holding the sprint during their annual conference made these logistics easy to arrange (although, as we'll explain shortly, there was some confusion over logistics at the beginning of the sprint).

Publicity also turned out to be easy because the FSF has a large following. Because the topic is so thoroughly understood by free software users, many were happy to sign up for a chapter or more. Publicity consisted of:

- The normal announcement on the FLOSS Manuals site, plus a blog by Adam.
- An FSF press release that mentioned the sprint in conjunction with their agreement to form a long-term relationship with FLOSS Manuals.
- Another FSF announcement to members, inviting them to participate.
- Two blogs by Andy on the O'Reilly web site.

Some of these were picked up by outside observers, who endorsed the project in advance and gave it welcome publicity.

As he was living in the area, Andy tried to recruit personal contacts and colleagues at work. (He volunteered to do the coordination for the project on the FLOSS Manuals because he works in Cambridge. As it turned out, Adam came into town to lead the sprint and did more coordination.) Although Andy received some supportive responses from friends and colleagues, none of these people ended up attending.

Before striking a deal with FSF, Adam visited their Cambridge, Massachusetts office a couple times. Andy visited during the week before the sprint.

Andy's visit to the FSF office was definitely useful. Meeting some FSF staff he hadn't met before helped to build trust on both sides and make us work more as a unit. We also determined a couple last-minute needs for the sprint.

Andy also performed another unusual task before the sprint: he wrote an early chapter. His goal was to provide a model that would hopefully encourage other authors to meet the goals Andy set for the project:

- To keep in mind at all times that readers might be technical novices
- As a result, to move very slowly from topic to topic, explaining each new concept thoroughly
- To maintain a positive and encouraging tone that reminds the readers regularly that these topics are within their grasp
- To use many examples

Although each FLOSS Manual has a chapter on writing guidelines, Andy felt that something more concrete was needed to convey the goals to authors; hence the sample chapter.

Most of the submissions met these goals, although we can't determine whether providing a sample chapter helped to create the environment for this success.

Set-up

Several snafus slowed us down on Saturday morning when the sprint started: it was hard to figure out which rooms were ours, a class was being held in one room that had been promised to us, we had to put up directions to the rooms and had prepared only some crude signs with 8x11 sheets and masking tape, etc. None of these ended up being much of a problem.

Certainly, it's important to know the space beforehand (Andy at least had spent time in the facility), and it would have been helpful to plan more and prepare better signage.

We were also a bit worried at the start because only one person joined us. A couple others who came in later were willing to help but hadn't brought laptops. Andy offered his laptop out of politeness, but ultimately they decided they couldn't be productive.



But it was obvious from the start that many remote authors were busy writing, so Adam and Andy just had a good time writing, editing, and chatting over the IM interface of FLOSS Manuals. More participants filtered into the room.

Bandwidth, luckily, was ample on the wireless network, and participants on-site could use an account that the host venue (Harvard University) provided to the FSF conference.

Adam brought snacks to supplement the conference food (which was quite good).

Process

On the second day, several more people came to write. There were usually about four people in the room at any one time. Andy came at the beginning of each day, stayed for four or five hours, and logged in later from home. Adam was on-site the whole time. The number of remote contributors remained high.

Andy tried to focus on editing submissions, although his intrinsic love of writing often got the better of him and he would spend half an hour or so writing new material. Andy looked over the outline and existing chapters from time to time in order to find sections that needed to be written. Luckily, submissions came in so fast that we had to explicitly ask for help on sections only a couple times.

The manual was over one hundred and sixty pages in size at the end of the day Sunday, and all the sections except for two non-essential ones were considered ready for publication. Adam pushed the button and the manual went up for viewing and for sale. It was judged quite good by the FSF and others, and quickly made it onto a list of recommended free manuals by one third-party web site.

Post-sprint tasks

A few days after the sprint, both Adam and Andy looked over the whole manual. A couple other people, notably FLOSS Manuals member Ed Cherlin, also read it over and made contributions.

Ed wrote a good deal of a glossary, which had not received much attention during the sprint. He also made some suggestions and did some rewriting.

Andy harmonized the style in many places, removed a good deal of redundancy, and wrote a couple sections to flesh out one chapter.

Adam determined sections that should be reordered, so that simple material came earlier in the book than complex material. Given the goals of "friendliness" that Andy defined for the book at the beginning, this was an important task.

Evaluation

The sprint was viewed by both FLOSS Manuals and FSF as an important indicator of whether we could work together. Both sides were very happy with the outcome.

Andy, who had originally volunteered because he was geographically in the area of the sprint, took on a leadership role as a proof of his ability to contribute professionally. As it turned out, the FSF handled the logistics, so Andy contributed by writing the outline and sample chapter, blogging about the event before and afterward, editing text during and after the sprint, and writing many sections. The amount of time he put in was definitely related to his desire to prove himself on his first sprint.

As mentioned earlier, Andy was concerned with making the manual very easy for novices, particularly people with a bit of a phobia about the command-line and related tools (such as text editors). It seemed like the book needed this extreme attention to usability in order to meet the needs of its audience. Andy also considered that many publishers had released well-regarded introductions to the tools in the book, which created high standards this book should meet.

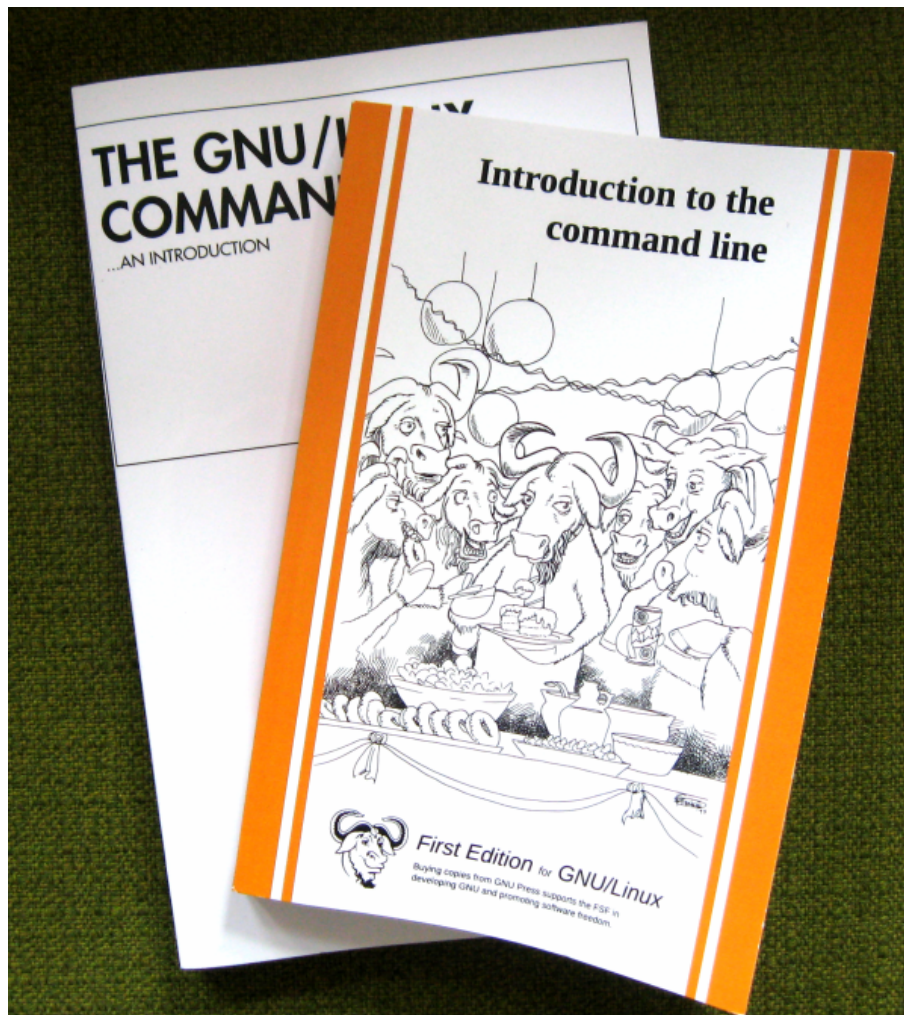
As mentioned earlier, the vast majority of the authors picked up (from Andy, from each other, or just from common sense) the need for a slow pace and simple explanations. Thus, editing on most sections went quickly, although Andy and others would add new material as they went along.

It's possible that, as FLOSS Manuals becomes more well-known, sprints will increasingly involve large numbers of remote contributors as this one did. It's good to know that the book sprint process and the FLOSS Manuals web site can handle the flood. It would seem likely that, under these circumstances, the steps we took to prepare for the sprint contributed to its success:

- Provide a detailed outline
- Divide the project into many small web pages to facilitate concurrent authors
- Provide a sample chapter
- Edit early submissions quickly to maintain the goals of the project consistently

Outcomes

We now have two beautiful books from this sprint - both with the same content. The Free Software Foundation released their own version of the manual with their own cover and are selling it from their site.



CiviCRM

This book sprint was a week-long event, preceded by considerable planning, that took place in a resort area near Lake Tahoe, California from May 3-8, 2009. This sprint resulted in a book of approximately 280 pages that was highly regarded by the community.



The CiviCRM project and community

CiviCRM is a free CRM projects. There are many free software projects in this space, but CiviCRM is distinguished by its focus on non-profit, government, and advocacy sectors. They prefer to use the term Constituent Relationship Management for CRM instead of the common expansion, Customer Relationship Management.

CiviCRM's documentation is typical for a free project, though probably somewhat superior. They have tutorials and examples, along with complete reference documentation. Community forums are active. One factor that put their educational resources a notch above most projects is a collection of test sites where you can practice your skills as a CiviCRM administrator without first installing it.

Nevertheless, the CiviCRM leaders felt a need for improving their documentation. They decided to hold a sprint in San Francisco and to fly Adam in to lead it. They also agree to pay Andy to provide organizational and editorial support.

Platform Development

For this sprint Aco developed a feedback mechanism which would send notifications to the FLOSS manuals web chat when someone started editing a chapter. The feedback is italicised and looks like this :

AdamHyde is editing BookSprints - FLOSSify.

AdamHyde is editing BookSprints - CaseStudyFour.

This has actually proven more useful for non-sprint days as it is possible in real time to see who is active in FLOSS Manuals editing manuals other than the one you are working on.

Organization

The division of labor between Andy and Adam went well, except for one issue that fell through the cracks, as we'll see. Adam, who has had more experience than anyone leading book sprints (although he still feels he's experimenting a lot, and may always experiment a lot), explained the process to everyone and came to San Francisco to be Sprint Master. Andy did not travel to San Francisco, but put in over fifteen hours beforehand mobilizing the community and leading discussions related to content. Andy also did substantial editing during the sprint.

As with most sprints, pre-planning was concerned with three tasks (aside from logistics such as food and accommodations):

- Determining the scope and topics. This proved more difficult than with most sprints.
- Developing the outline, which required recruiting and motivating subject matter experts.
- Recruiting authors.

The pre-planning started with productive discussion on a mailing list set up by CiviCRM explicitly for the sprint. Members were added to the list by invitation. In addition to Adam and Andy, it contained about a dozen people with extensive experience deploying CiviCRM in their own organizations and as consultants. They had tremendous experience and lots of ideas about what's missing from the documentation. Their suggestions were deep: how to prepare your organization, how to map your current practice onto CiviCRM, etc.

However, the group needed coordination. There was too much centrifugal force during early discussions; nothing was coalescing. The mailing list members discussed whether the audience should be end-users, administrators, advanced programmers, or outsiders interested in learning what CiviCRM had to offer. They considered nitty-gritty administration books (particularly on advanced topics) and very high-level, conceptual books. They debated different ways to incorporate case studies.

All these ideas were good; the question was which to do for this sprint. Andy kept them talking, told them to write down ideas, and tried to formalize what they wrote so it looked actionable.

Eventually Andy realized we should have a vision statement. This might sound like overkill for a sprint, but it seemed useful as a way to anchor the group and give them a guide so they could prioritize. Andy summarized the trend of the conversation, but there was no real work on the vision statement and it didn't prove useful when a quick decision was needed later.

A teleconference lasting more than one hour was also held about ten days before the sprint. Half a dozen CiviCRM leaders joined in, along with Andy (Adam, unfortunately, was not present) and a great deal of progress was made in defining CiviCRM's needs and dividing up tasks before the sprint.

Up until that time, project members were reluctant to add to the outline. Ideas flew back and forth on the mailing list, and Andy added what he could derive from this discussion to the outline page, but it wasn't developing. In fact, it took some effort just to persuade project members to sign up for FLOSS Manuals accounts so they could try out the software and edit the outline. Some people submitted stubs for case studies, but there was little material in them.

About two weeks before the start of the sprint, the team was uncertain of what to do and whether they were going in the right direction. Adam had to reassure them that things are normally open-ended and that the sprint itself is the only place to resolve a lot of the issues.

But thanks to the mailing list discussions and especially the teleconference, momentum picked up. A week after the teleconference, the outline was not only quite large but well organized. There seemed to be a sense of "we're cookin'" and "we can do it."

It should also be noted that, even though we spent much more time planning the sprint than most projects do, some participants wished we had done more.

Controversy over outline

So in one week we went from too little to too much. Having participated in only one book sprint (an unusual one, the Command Line book) Andy didn't recognize that there were several times too many topics in the outline to write about in one week. About a week out from the sprint, Adam recognized the problem and decided to cut the outline way down.

In retrospect, we couldn't see any way to hold back contributions. Andy needed to take strong steps during the weeks before the sprint to encourage contributions to the outline. Once you open that spigot, you can't turn it off without sounding discouraging.

As mentioned, Adam had actively decided not to participate in detailed discussion of the outline and leave this to Andy--although Adam was on the mailing list--and was not part of the teleconference. Andy had been intensely involved, but didn't have enough experience to size the outline.

Adam presented the truncated outline to some CiviCRM community leaders the week before via email and at a real space meeting, however they rejected it.

Adams approach was then to re-open the discussion about the outline on the first day that everyone arrived at the sprint. The sprint team quickly created an outline, with Adam playing a facilitator role, so we got over that mini-crisis. Interestingly the outline created by the CiviCRM team on site at the sprint reflected almost exactly the outline Adam had presented a few days before.

Sprint-week editing

The team worked very hard during the week, often logging in early and staying late. Like all sprints, the level of professionalism in the actual writing varied, but the ideas were almost 100% on target (and the things that were inappropriate for the book got weeded out).



Andy edited a few chapters very early in the week to set professional expectations for pace and style. Most of his questions were along the lines of: "does the audience need this?" and "what's missing?" Andy also put in most of the day Wednesday on editing.

By Friday, the team decided to concentrate on filling in gaps and polishing what they had done the rest of the week. The finished at 6:00 PM and published the book. 260 pages - our biggest yet.

Fun

During the week we also managed to get out a bit and have some fun!



Follow-up

The book was considered a success by CiviCRM leaders. The usual clean-up was performed in the following few weeks. Two sprinters noticed that some of their material had been removed late in the sprint. There is still discussion about where this material could fit in to the book. We believe a long-term maintainer has also been found.

In addition, the FLOSS Manuals editing interface has been slightly simplified to avoid the temptation by contributors to over-complicate the layout which occurred a few times during this sprint. The layout has to be kept simple so the material is rendered well in both html and pdf formats. This will be less of a problem in the future as FLOSS manuals has developed a more complicated PDF rendering engine that supports the full spectrum of CSS layout, so a more complicated WYSIWYG layout may be an option in the near future.

Feedback

Some feedback sent in by the CiviCRM Sprint participants after the book had been out for a few weeks :

"I received the following feedback from a potential CiviCRM user who was reading our book. She had some basic questions and she used the Organization information in the Examples section to send an email to the AFSC's general information address. That email was forwarded to me.

After I answered her questions, giving CiviCRM glowing reviews of course J I asked her what she thought

about the book. This is what she said:

'And I would like to say, if by book you mean the website <http://en.flossmanuals.net/CiviCRM>, I thought it was very helpful and really well organized so wonderful job there'.

When I read her comment I immediately thought of all of you and wanted you to be a part of what I was feeling, which was pretty proud of all the work that we did. It feels really good to know that people are actually referencing it and that at least for some, it is helpful. "

Cynthia Tarascio, Book Sprint Participant

=====

"I've also gotten some very positive feedback from folks on IRC. We are frequently referring folks to it on the forum when it's clear they need "the big picture".

Dave Greenberg, CiviCRM founder and sprint participant

=====

"I actually received some feedback recently that Iâ ve been meaning to send out to the group. I was contacted by someone who was in the process of implementing Civi and needed some assistance (heâ s already begun implementation). He mentioned having read significant portions of the book, and was particularly impressed with some of the early chapters that dealt with project management, identifying key players, etc. He said he took a step back to rethink how he approached some aspects of the project after reading those sections."

Brian Shaughnessy, Book Sprint Participant

=====

"I met someone from Greenet recently (an ethical UK ISP) at a London Net Tuesday event. He was a developer who was doing his first development with a client and said he found the book super useful in getting to know the system. He said that he learnt loads about Civi through the book and was really grateful to have access to it.

I've also seen 3 copies of it out and about on my travels - and watched 2 people buy it online (and I didn't pressure them into buying it - honest!)"

Michael McAndrew, Book Sprint Participant and Manual Maintainer

=====

"I met with the IT manager of a (new zealand) political party today who was considering CiviCRM. He contacted me through the Circus Trust as he had read the case study.

Anyway, he was really positive about the book and made the comment that 'IT manuals normally put him to sleep but this one was interesting and well-written'. It was obvious that reading the book was having a significant influence on the likelihood of him going down the CiviCRM path and also increase the amount of functionality he would take advantage of.

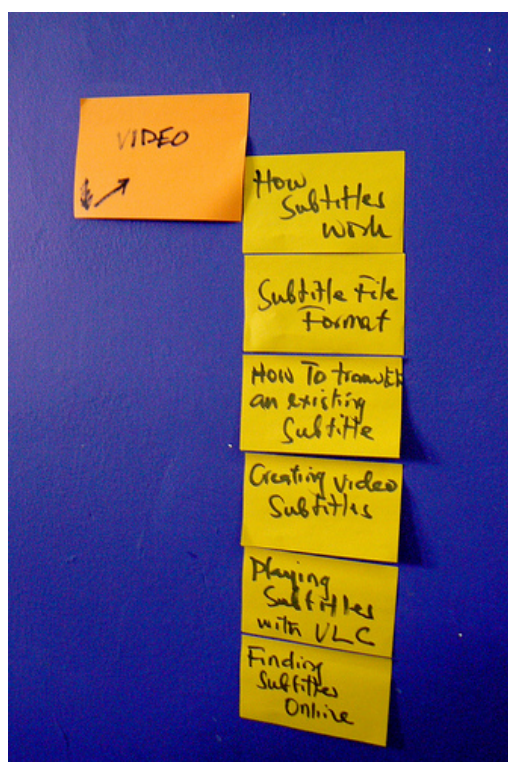
So, nice to see that not only do people appreciate the book but that it also achieves it's goal of getting people to use Civi & to use it better."

Eileen McNaughton, Book Sprint Participant

Open Translation Tools Sprint

Once again, the Book Sprint proves there are no fixed rules. This sprint evolved out of the Open Translation Tools conferences that have been organised by San Francisco based Aspiration (<http://www.aspirationtech.org>). From the outset we were determined to find a way to mix the OTT conference, facilitated by Allen Gunn, into the pre-sprint development.

We had no clear idea about how this was to develop, except that the sprint facilitator (Adam Hyde) would be present at each session to show people the FLOSS Manuals tool set and how to use it, and explain to anyone that wanted to know the idea of a Book Sprint. One idea which was discussed in brief was to write out a suggested Index on paper and put it on the wall for discussion. Building on a suggestion from the CiviCRM sprint (to have "more sticky notes") Adam built the Index on the wall using sticky notes with two of the unconference participants (thanks Eileen and Peter for this suggestion!).



Through out the next three days participants were invited to comment on the index, replace and add items, and commit to writing material remotely (after the conference had finished).

This process was quite effective in drawing people together to discuss the sprint, and it did result in some remote contributions, but this process could probably be more effectively utilised by getting specific commitments from people, possibly noted in a spread sheet or simple text document.

The Venue

We were situated at De Waag, a beautiful venue at the heart of Amsterdam. The venue was sponsored by The Society for Old and New Media which is located in the building.



It was a great venue for the event and added a very unique 'once in a life time' feel to the experience.

The Sprint

The unconference ran for 3 days, and then hot on the heels was 5 days of sprints. We were suffering a little from fatigue but still there was a lot done on the first day. The dynamic of the sprint was different from previous sprints in that most of the people with 'expert' knowledge were only in the sprint for the first day. We had 15 people on that day, and only 5-9 in the following days (most of which did not know a great deal about the topic). This low 'expert' to 'newbie' ratio was a product of the time commitment required to do a 3 day conference, and then a 5 day sprint. Not many people can commit to that kind of time. So we had to work in a slightly different manner. Firstly, day one was not a discussion day about the Index or scope of the manual, it was just writing. The idea was to get as much content down as possible without discussion on the scope, target audience, tone etc. The next days were spent 'ironing out' the content that was dumped that day.



This worked very effectively. Day two the smaller team spent the day writing, and structuring the first two sections. By 1/2 way through the third day, these sections were completed. The rest of days 3 and 4 were spent restructuring the next 4 sections, and the last day was committed to cleaning up and filling out the existing content in the later sections. This got us home.

Interestingly, there was no discussion about target audience. However this became clear as the group rolled down the table of contents and realised that the manual was pretty much targeted at people wanting to learn about Open Translation and get involved. Hence the manual was an overview coupled with some specifics about how to get involved and how to improve your translations.

Book Two

In day two it was apparent that there were 3 contributors that did not have much expertise in translation but who could contribute a lot to another manual on video subtitling. So this group split off from the other sprinters and focused on writing a very practical work book for subtitling video using Free Software. Since FLOSS Manuals already had some manuals available on subtitling tools, this material could be included in the manual using our 'include' mechanism.

This second book is very good and totaled 120 pages of very useful and practical information for anyone wanting to make subtitles.

Facilitator Contributor

One interesting issue that arose is that the facilitator (Adam) also contributed a lot of content. This is not usually the case as the facilitator is usually facilitating. However, due to the low number of people with 'domain knowledge' the facilitator wrote a lot of material. This led to a confusion at one point whereby a participant was frustrated because the facilitator was deeply involved in some tone and content discussions and speaking as a contributor.

How this might be resolved in future is an interesting question. Is it best for the facilitator to be involved only in the role of facilitation or is it appropriate for the facilitator to contribute content as well? The later strategy might become an issue if the facilitator has a position on the content they are writing which is in conflict with other members of the team. Who is there to mediate this if the facilitator is 'inside' the contributor circle?

Printing Out

In this Sprint we printed out the chapters far more often than previous sprints.



This was a very effective way of visualising the content and working our way down the table of contents, restructuring and smoothing as we went.

License

All chapters copyright of the authors (see below). Unless otherwise stated all chapters in this manual licensed with **GNU General Public License version 2**

This documentation is free documentation; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This documentation is distributed in the hope that it will be useful, but **WITHOUT ANY WARRANTY**; without even the implied warranty of **MERCHANTABILITY** or **FITNESS FOR A PARTICULAR PURPOSE**. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this documentation; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Authors

ADAMS TIPS

© adam hyde 2009

THE AIMS

© adam hyde 2008, 2009

Modifications:

Andy Oram 2009

ANNES TIPS

© adam hyde 2008, 2009

Modifications:

Anne Gentle 2008, 2009

BUDGET

© adam hyde 2008, 2009

Modifications:

Andy Oram 2009

Anne Gentle 2009

COMMAND LINE

© adam hyde 2009

Modifications:

Andy Oram 2009

INKSCAPE

© adam hyde 2008, 2009

XO / SUGAR

© Janet Swisher And Anne Gentle 2008

Modifications:

adam hyde 2009

Anne Gentle 2008

CIVICRM

© Andy Oram 2009

Modifications:

adam hyde 2009

CREDITS

© adam hyde 2006, 2007, 2008

FLOSSIFY ONE

© adam hyde 2009

COMMUNITIES

© adam hyde 2008, 2009

Modifications:

Andy Oram 2009

INTRODUCTION

© adam hyde 2006, 2007, 2009

Modifications:

Andy Oram 2009

Anne Gentle 2008, 2009

OTT

© adam hyde 2009

THE PLAN

© adam hyde 2008, 2009

Modifications:

Andy Oram 2009

Anne Gentle 2008

Austin Martin 2009

THE PROCESS

© adam hyde 2009

Modifications:

Andy Oram 2009

REMOTE CONTRIBUTIONS

© adam hyde 2008, 2009

Modifications:

Andy Oram 2009

ROLES

© adam hyde 2008, 2009

Modifications:

Andy Oram 2009

Anne Gentle 2008

TIMELINE

© adam hyde 2008, 2009

Modifications:

Andy Oram 2009

Anne Gentle 2008, 2009

TOOLS

© adam hyde 2008, 2009

Modifications:

Andy Oram 2009

Anne Gentle 2008

VENUE

© adam hyde 2008, 2009

Modifications:

Andy Oram 2009

General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without

limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a)** You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b)** You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c)** If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a)** Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b)** Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c)** Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

- 4.** You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
- 5.** You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
- 6.** Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
- 7.** If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS