[ZenDoodles]

Our goals:

1. Introduce contribution training resources
2. Introduce all the stuff happening at *this* event, and prep you (yes you!) to participate.
3. Give you the resources to plan a contribution sprint or training at your own user group, company, or Drupal event.

[order: ZenDoodles, xjm, add1son, YesCT]
We each introduce ourselves.
photos, first names, user names, and companies

[YesCT] I am Cathy Theys, YesCT on drupal.org. I work at comm-press based in Hamburg Germany doing community contribution and social media. I am a regular mentor for the weekly core office hours in irc. I've organized the Global Sprint Weekend that was held in March.

We've led nine live core mentoring sprints at Drupal events in the past year.
Also, if you look around the room, everyone wearing a green Portland shirt is going to help with the sprint here at Portland.

[ZenDoodles] Session outline (or maybe on the next slide? whatever ZenDoodles wants)
What is a sprint?
The case for core contribution
Contribution solutions
Planning a sprint
How to be an effective mentor
What's next?

Remember!

- Use contribution mentoring tools
- Prepare a task list in advance
- Plan for development environment setup

[ZenDoodles]
If you remember nothing else from this session, remember these things

Zendoodles reworking (nudging) a three panel slide to reuse here that reuses images, sliced, from the three main points. Also probably take ‘Remember’ and use the key words we want them to remember.

What is a sprint?

[ZenDoodles]

A sprint is a focused contribution event

[ZenDoodles]

Some sprinters want training

[ZenDoodles]

Some sprinters want coaches

[ZenDoodles]

The case for core contribution

[xjm]

So I'm going to talk a bit about the case for core contribution, or the problems we're trying to solve.

I'd like to begin by telling the story of how I got involved, because I think I faced the same challenges a lot of people do.

Core is the best way to learn

[xjm]

Two years ago, I was trying to upgrade the Taxonomy Access Control module to Drupal 7 for a client. I knew nothing about Drupal 7 at the start, but with some help I was able to get about 80% of the way through it before I got stuck. I talked to some other people about it, and we came to the conclusion that we needed a new alter hook. hook_field_widget_form_alter(). So we're like, okay, we're going to do this, we're going file a Drupal core issue asking for this hook. (gesture)

So we posted the issue, and... I kid you not, sun had rolled a patch for this two hours later, like I came back from lunch and there was a patch there. But it was failing tests, and I didn't know how to fix it. By this point my port of TAC is a couple weeks late for my contract, and so I started thinking, you know, what can I do to make this patch get in faster? And people suggested working on other core issues. My reaction is, okay, so, I'm not a core developer; I don't know what you expect me to do. I don't know anything about this stuff.

As fate would have it, two things happened that same week.

The first was that chx asked in IRC for someone to update a patch, just taking out an extra blank line that he'd accidentally added. And I go, huh. I could totally do THAT. So I did it. I downloaded the patch, applied it, made this one small change, and then uploaded a new patch.

And it was such a tiny, silly thing. chx could of course have made the change himself in like thirty second flat. But instead of doing that, he gave me the opportunity to take that first step, to cross that line in my head.

The second thing that happened that same week is that the issue summary functionality was deployed on Drupal.org. For anyone who's not familiar, the main post of a Drupal.org issue is editable. The idea is that anyone who's looking at an issue can collaborate to update it, to describe the current status. So instead of having to read through fifty or 100 comments to figure out what's going on, you can just read the summary.

So I started writing issue summaries. Each day, I took one Drupal core issue, read the whole thing, researched things in the issue I didn't understand, and then wrote up a concise, up-to-date summary to save everyone else time.

Eventually the patch for hook_field_widget_form_alter() was RTBCed, and was just stuck near the end of a big backlog of RTBC issues. RTBC by the way stands for "Reviewed and Tested by the Community," meaning it's ready for a core maintainer to look at it. So I figured that by saving the core maintainers time reading these issues, I'd also help myself in the long run, because the less time they had to spend making heads or tails of a patch, the sooner they'd get to mine.

A bunch of stuff happened as a result of me writing those summaries, beyond helping my patch get in. One thing was that every now and then I'd come across an issue that I actually knew how to fix. And thanks to the encouragement chx had given me, I was more comfortable with making patches for those issues myself.

Another thing is that I figured out that there were a lot other Drupalists in the same situation as me, who wished they could contribute to these issues, but just didn't know where to start. So I tried to get other people to help write issue summaries too, and that ended up being one of the seeds for me for the core mentoring initiative.

And finally, completely by accident, I taught myself about core. By reading an issue every day, researching what I didn't understand, and then trying to explain it to other people, I learned a LOT. I started writing automated tests for bugs, and fixing more and more of those bugs myself. I started reviewing other people's code. And it was so entirely different from working on a contributed module. Instead of just hacking on things until they worked, I learned how to do them the right ways.

So that's why I think core is the best way to learn.

1. You learn a lot about Drupal
2. You get more feedback and interact with more people than in contrib
3. You're helping others at the same time
4. And you get this ownership in Drupal, this empowerment to solve problems you encounter.

[xjm]
So let's look a bit at who contributes to Drupal core.
This graph shows the number of developers contributing to core for the past five major releases.
As you can see, the number increases steadily,
from about 300 in Drupal 4.7 to over 1200 already for Drupal 8--this graph is from a month ago--and Drupal 8's not even done yet.
Lots of people to learn from.

[xjm]

Something to keep in mind, though, is that Drupal's adoption is increasing faster than the number of core contributors.
As a percentage of the active users on Drupal.org, the number of core developers is actually dropping.
This is counting only active Drupal.org accounts,
people that logged in at least once during the year each version was released.
And we know that only a fraction of Drupal users even have Drupal.org accounts.

The workload isn't evenly distributed, either; most of these 1200 developers are what we call the "long tail." About half of the 1200 people have contributed to just one patch, while the top dozen or so contributors have contributed to hundreds of issues each.

[xjm]
And finally, there's a large burden on existing regular contributors.
Right now there's nearly 7000 issues filed against Drupal 8 alone. This is as of the end of April.

Core affects all of us, right?
If we rely on only a few hundred people to do the work on most of those issues, that's clearly a problem.

Nearly half of these issues are active, meaning there's no patch in the issue yet.
Over five hundred of them have zero replies.

And almost 20% of them are sitting at needs review.
One thing that's an ongoing challenge in Drupal core is a shortage of reviewers.
Ideally that slice would be much, much smaller
most of those issues should be marked needs work, and some might be RTBC
but a lot of them probably don't even apply anymore because they've been waiting for reviews for a month or longer

So there's a few things that need to happen, right?
We need to reduce the burden on this subset of core contributors.
We need to, ideally, recruit more core developers who will work on more issues.
We need to do something about the issues with no replies.
And finally, we especially need to something about the review gap, and make it easier to get good reviews faster.
(For anyone who's interested in that topic, by the way, Cathy and Andrea are doing a core conversation on patch reviews tomorrow.)

etup

[xjm]
It's hard, though, to grow the number of doers in core.
The first steps especially can be difficult, like they were for me.
Some of you have probably seen this picture before.
Some mock learning curves for different content management systems,
with Drupal's learning curve represented as an insurmountable cliff.
It's a joke, but it might resonate for anyone who tried to learn core development, especially a few years back.

There are, frankly speaking, barriers to core contribution.
1. The first, the most obvious, is this "I'm not smart enough" thing that I went through.
- Because of Drupal's famed complexity and flexibility,
- and maybe a bit because of our rockstar culture,
- we get the impression that there's nothing "little old me" can do.
2. Another thing is that core processes are hard to grok and more rigorous than contrib.
3. Also, it's unfortunately not entirely rare for people to have bad first experiences when they try to contribute,
- either unfriendly reviews, or worse, no reviews at all.
- This shortage of reviewers I mentioned means that someone's first ever patch
might go unlooked-at for months or years.
- That sucks, right? Who's going to try a second time when that happens?
4. Finally, people have difficulty finding tasks that they can do successfully.
- There's so many thousands of issues, so many things to learn, so much jargon.
- It's hard to know where to start.

Our goal in the core mentoring initiative is to give everyone a way up that cliff. Maybe even wear it down to a nice foothill. If you have Drupal site building experience, we want to make contributing back approachable to you.

Contribution solutions!

Help people find tasks to do and get the tools to do them

[xjm]
So, our solutions focus on helping people find the task they can do, and getting the tools to do them.

The tasks

[xjm]
In the core mentoring program, we've focused on helping people find tasks and guiding them through the contribution process.

One issue, many tasks

- Use contribution mentoring tools

View Edit Outline

Posted by xjm on *May 4, 2013 at 4:50pm*

Project:	Drupal core
Version:	8.x-dev
Component:	taxonomy.module
Category:	bug report
Priority:	normal
Assigned:	Unassigned
Status:	needs work
Issue tags:	needs accessibility review, Needs issue summary update, Needs manual testing, Needs tests

[xjm]
This was my big realization back in 2011:
One issue actually includes many tasks.
We don't get that far with one rockstar programmer fixing a bug by herself; core is collaborative.
You don't have to write a patch all by yourself to help an issue along.
It's not always obvious to people what these tasks are, though,
so to make it easier to find tasks we've introduced a set of issue tags that identify tasks that need to be done for an issue.
Needs manual testing, Needs reroll, Needs issue summary update, and so on.
And it'd be totally awesome, by the way, if everyone in this room got in the habit of adding those tags to issues when appropriate.

[xjm]

We also worked on some documentation in the Drupal.org handbook that explains how to do each of these tasks, to demystify the process.

I really recommend checking out this section of the handbook after the session is over.

It's at drupal.org slash contributor dash tasks.

[xjm]

Finally, for the core mentoring initiative, we picked out the task types that work the best during IRC mentoring and sprints, and grouped them by the tools you need to do them. This helps us figure out what we can start someone with based on their past experience and skillset.

The first group, the "A" tasks in core mentor parlance, just require a user account on Drupal.org. These include:

- writing summaries for issues
- documenting the steps to reproduce an issue
- and triaging a reported issue to verify it, make sure it's filed correctly, and look for duplicates.

It's important to keep in mind that these are not necessarily quick or easy tasks. Sometimes writing a good issue summary takes a lot of work, and you get more out of it if you put more in.

‘B’ tasks: Git

```
commit ae0984f8711db81fd2157d189bbddb1105a9cc19
Author: catch <catch@35733.no-reply.drupal.org>
Date:   Mon Apr 22 22:57:38 2013 +0100

    Issue #1942346 by swentel: Convert Field API variables to CMI.

commit 2d6ed6c862d3ce4e7a651f45108ead4f5ff6f2fd
Author: Alex Pott <alex.a.pott@googlemail.com>
Date:   Mon Apr 22 19:07:20 2013 +0100

    Issue #1955286 by droplet | nod_: JSHint translation_entity().
[tesla:drupal | Tue 08:41:18] $ git branch
  7.x
  8.x
  actions
* else_if
  field
  frontpage
[tesla:drupal | Tue 08:41:28] $ git rebase origin/8.x
Current branch else_if is up to date.
[tesla:drupal | Tue 08:41:36] $ git diff --stat origin/8.x
 core/modules/node/lib/Drupal/node/NodeStorageController.php | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
[tesla:drupal | Tue 08:41:50] $ git show
commit 10cedb1ac4513d3ac763b33dfb38fc5273b61282
Author: xjm <xjm@65776.no-reply.drupal.org>
Date:   Tue Apr 23 08:38:22 2013 -0500

    Correct else if to elseif in the node storage controller.

diff --git a/core/modules/node/lib/Drupal/node/NodeStorageController.php b/core/modules/node/lib/Drupal/node/NodeStorageContro
index c9e832e..0fc7921 100644
--- a/core/modules/node/lib/Drupal/node/NodeStorageController.php
+++ b/core/modules/node/lib/Drupal/node/NodeStorageController.php
@@ -103,7 +103,7 @@ protected function invokeHook($hook, EntityInterface $entity) {
     if ($hook == 'insert' || $hook == 'update') {
       node_invoke($node, $hook);
     }
-    else if ($hook == 'predelete') {
+    elseif ($hook == 'predelete') {
       // 'delete' is triggered in 'predelete' is here to preserve hook ordering
       // from Drupal 7.
       node_invoke($node, 'delete');
```

[xjm]

The “B” tasks require using git, our version control system.
This includes small cleanups like that one-line fix chx asked me to do, as well as

- manually testing things
- creating documentation patches
- and rerolling patches so that they apply to the latest HEAD

‘C’ tasks: Programming

[xjm]

And finally, the “C” group of tasks require programming skill.

- writing patches,
- writing automated tests,
- doing code reviews,
- and writing API change notifications.

[xjm]

We also built a website to assign people to tasks, because the Drupal.org issue queues don't really do what we need them to.
The site is drupalofficehours.org.
We it to coordinate both sprints and core mentoring in IRC.
It provides a curated list of tasks for core issues.
Having this site

1. Filters the noise of the core queue
2. Helps match contributor skills to available tasks
3. Allows us to target important issues
4. And, most importantly, matches contributors with mentors
    1. build community
    2. task selection (self-assessment)
    3. help through the process, a place to go for questions
    4. the mentor doesn't need to know all the answers, just who to ask
    5. closing the circuit: task review. Feedback is essential.

We have a video explaining to use this site that you can watch after the session. It's at core dot drupalofficehours dot org slash videos.

[Addi]
Solution part B: Get people the tools they need to do the tasks

Community Tools Workshop

- [Insert slides for Addi here]

[Addi]

The Drupal Ladder

- Insert content here. But not the podcast.

[Addi]

Planning a sprint

[YesCT]
I'm going to talk about what it takes to plan a sprint.
Think about: mentors, location, and publicity.
Select sprint tasks in advance,
and schedule a time after for doing task reviews.

Depending on the size of your sprint, there could be a lot more planning to do.
For example, the four of us have been working with the DA to plan this event since January. Smaller sprints need less planning.

Here are the essentials for when you plan your sprint.

[YesCT]
You can host a sprint and be the mentor.

[this image shows a large sprint room with many pairs leaning in, and gesturing to each other.]

Estimate the number of people that might attend.
A ratio of 1:8 keeps mentors busy, but not too busy. So if having more than 8 attendees, perhaps recruit mentors.

On the day of your sprint, if there are not enough mentors. No worries: pair experienced people with new people.

Working together is one of the best parts of a sprint. You dont have to do it all. When someone is having trouble with git, find someone just a bit more experienced. Ask them to help and move to sit next to the person having trouble. It builds the confidence of that on-the-spot new mentor to see they can help others.

Planning: Location

[YesCT]
At a minimum, make sure your location has:
wifi
outlet strips
food: people will leave if food is not there, and everyone will lose an hour and half of sprint time.
A simple way of dealing with food is to get a menu, order in.

When reserving a location for my local sprint during the March SprintWeekend, I asked about wifi (they had it), outlet strips (they didn't. so we would bring some). I found a sponsor for food. But what I didn't ask about was tables. It was a meeting room that doubled as a yoga studio. And, the day before the sprint, I was picking up the key, and they asked if we would like tables. !!! I said yes. :) Phew! I'm glad they asked. Be sure to confirm your space has what you need. This is a picture us at 2 tables with lunch evident. We ordered in lunch (only about $80 to feed us all).

Round tables work best, or with people on each side of rectangular tables facing each other.

[YesCT]
For publicity, make an event on groups.drupal.org. That makes it show on the cool drupical.com map. This image is a sample of what the drupalical map looks like with pins, marking events all over the world. Picture your event pin here!

Many people also make an event on meetup.com

Announce who is welcome at the sprint, usually that is: anyone with site building experience, and also more experienced contributors.
Remind attendees to bring... what you need them to bring:
 laptop, outlet strip, snacks?

Planning:
Development environments

[YesCT]
On the day, people will have trouble with their development environments running drupal 8. Plan for this.

This close-up of laptops at a sprint pictures Macs, other environments are equally popular.

Find people who have experience with setting up each: Linux, Mac, Windows, ask them to come to your sprint. Encourage people to set up their dev environments before the sprint day. Send a reminder for them to do that at home, include resources that help them set up their environment.

Some tasks can be done without a dev environment, when doing your task selection, select a range of those. Also, simplytest.me rocks. And help people attending your sprint do tasks, even without a local setup.

Selecting sprint tasks

A: Drupal.org

Name	Where to find issues	Instructions
Issue summary	Issues tagged with "Needs issue summary update"	Write an issue summary
Steps to reproduce	Issues tagged with "Needs steps to reproduce"	Document STR
Triage and find duplicates	Issues with 0 replies	Verify a reported issue

B: Git

Name	Where to find issues	Instructions
Improve patch docs/code style	Issues marked CNR or CNW	Improve patch coding style and documentation
Manual testing	Issues tagged with "Needs manual testing" or "Needs screenshot"	Manually test a patch for a Drupal issue
Reroll	Issues tagged with "Needs reroll"	Re-rolling patches
Backport	Issues with status "Patch (to be ported)	Backport a Drupal Core patch
Create documentation patch	Documentation issues	Create a Drupal core patch

C: Programmer

Name	Where to find issues	Instructions
Patch review	Issues marked CNR	Review a Drupal core patch
Write tests	Issues tagged with "Needs tests"	Write a Drupal core automated test
Create patch	Novice issues	Create a Drupal core patch
Draft change notification	Issues tagged with "Needs change notification"	Write a change record

http://core.drupalofficehours.org/task-info

[YesCT]
Selecting the tasks is the bare minimum planning you should do ahead of time.
It is essential for a smooth sprint.
Select tasks no more than three days in advance. If too far in advance, the tasks might not be relevant by the time of your sprint.

This table shows the core office hours site task info table, it's a good resource for you for later. Jess talked about the task types, be sure to pick a few of each tool set: Drupal.org, git, coding.

We have a link to resources to help you with planning your sprint at the end of the session.
http://core.drupalofficehours.org/sprint
Part of http://core.drupalofficehours.org/lead-a-sprint

Sprint day

[YesCT]

This photo is typical of what gets posted at a sprint, it has a todo list.

. create account on d.o
. some way to take attendance (this one asks people to add their d.o username to a tiny url to a google doc)
. it tells people to install a web stack
. to install git
. it tells people clone and install d8
. it has a way to pick a issue to work on
. and how to deal with coordination (assign issue to themself, tag with their sprint tag)
. it also has the wireless password

In general, on the day, have signage so people know they are in the right location.
Welcome each person who comes, and introduce yourself.
Post a link to instructions for them (either a doc you wrote up or one from our resources),
and inlcude the wifi and password.
Get to know the attendees, asking them questions, and then assign tasks that are good matches to them.
Take pictures, write down who is there.

After the sprint: Task review

Tasks for mentor review

All tasks below need a review from a mentor to ensure they were completed correctly. See the reviewer instructions.

Task level: - Any - | Task type: - Any - | Issue priority: - Any - | Quick review?: - Any - | Apply

Task	Participant	Mentor	Drupal.org issue	Priority	Status	Version
Create sub-issues for router conversions	jberges	valthebald	1971384: [META] Convert page callbacks to controllers	major	active	8.x-dev
Improve patch docs/code style	friveroll	disasm	1971206: Cleanup of Book Manager Service	normal	active	8.x-dev
Update patch	shanethehat	xjm	1939082: Convert theme_status_messages() to Twig	normal	needs work	8.x-dev
Manual testing	leochid	shyamala	1847314: Reduce the dependency on javascript for the toolbar to display properly	normal	needs review	8.x-dev
Issue summary	kstonef	Kay V	1007746: Reordering fails with more than 100 items in a menu	major	needs work	8.x-dev
Steps to reproduce	no_angel	YesCT	1945226: Add language selector on menus	normal	active	8.x-dev
Issue summary	kstonef	Kay V	1007746: Reordering fails with more than 100 items in a menu	major	needs work	8.x-dev
Triage and find duplicates	capuleto	Cottser	1912474: Unable to pre-populate db in settings.php	normal	active	8.x-dev
Backport	chrisjlee	Cottser	1261846: Document 1MB maximum size limit for cache_set()	normal	needs work	7.x-dev
Issue summary	no_angel	YesCT	1912806: [illegible] page	[illegible]	[illegible]	[illegible]

http://core.drupalofficehours.org/tasks-review

[YesCT]

Feedback is really important to making new contributors stick around.
For someone new, getting timely feedback makes them feel like their work is not going into a black hole.
Also, remember the pie chart Jess talked about of all those issues? Reviews help move issues along.
Often the mentors make a special effort to either do reviews of tasks that happened during their sprint, or recruit people to help do the reviews.
Here is a screenshot of the tasks review page on the core office hours site that is handy if task selection was done using that tool. Otherwise you'll probably have a spreadsheet of tasks your participants worked on. Or, if they tagged tasks on drupal.org with your sprint tag, you can use d.o to generate a list of tasks from your sprint that need review.

For our Get Involved sprint on Friday, we have a review fest scheduled for one week later, June 1st 4-8pm PST meeting in irc in #drupal-contribute. We'll tweet about it. :)

[optional additional info: might be too much:]
When doing reviews, thank them for what they did, be specific.
Also, give them resources for what they need to know to fix something.
And explicitly state what the next steps are.

[ZenDoodles]

We'll go over everything that's happening on sprint day

This is going to be the biggest sprint day ever, so if you help out with this sprint, you're in great shape for doing your own sprint as well

The packet also has resources and suggestions for smaller sprints

Portland Core Mentoring

Tues., May 21	**3:15p:** This session **4:15p:** Task entry BoF I
Wed., May 22	**10:45a:** Patch Reviews **11:15a:** New contributors & core initiatives **1:00p:** Upgrading modules to Drupal 8 **2:15p:** Development environment BoF
Thurs., May 23	**10:45a-1:00p:** Task entry BoF II **2:15p:** Dries Drupal 8 Q&A
Fri., May 24	**9:00a-12:00p:** Workshop **9:00a-4:00p:** Mentored sprint **7:00p:** Mentor dinner!
Sat., June 1	**6:00p-10:00p:** Review fest in IRC

[ZenDoodles]

Also hand out the postcards with this info

Three parts, three audiences

[ZenDoodles]

image: map of sprint rooms

First two parts are our parts

What roles mentors can play (see also http://portland2013.drupal.org/volunteer/sprints)

1. morning and afternoon shifts
2. helping with training
3. helping with dev stacks
4. helping participants select and work on tasks
5. reviewing tasks during and after the sprint
6. (also, LFM for BoFs)

Morning intro

[ZenDoodles]

Simultaneous workshop & mentored sprint

[ZenDoodles]

Sprinters work in pairs, mentors help

[ZenDoodles]
Needs visual
This is important
walk around and help people
ask what they are working on and how it is going

After the workshop: dev environment fun time

[ZenDoodles]

[ZenDoodles]

Once people are settled in (and mostly set up with D8/git/etc.), it can be good to take "breaks" to re-engage people and demonstrate things like patch creation, patch review, etc. on the large screen. (This plays better once people have had a chance to "get their hands dirty" a bit.) At Denver and Munich we also had live patch reviews and in each case core maintainers committed a contributor's first patch live, which was awesome.

[YesCT]
You know what to do to plan your own sprint, you know what's happening this week... here's how to do it well.

Collaborate

[YesCT]

Collaborate. This photo from Munich shows one of the best things at a sprint, two people sitting next to each other at a table, working together!

When I was at Drupalcon Denver, I was not planning the sprint, or even a mentor (yet). But I had trouble with drush in my local dev environment, and so did someone else near me. So we decided to sit at the same table and work on it together. Then, we thought others must be having the same problem too, so we asked if the mentors would announce on the speaker that anyone with trouble with drush come sit at our table: the getting drush to work table. It worked, regular people collaborating to solve a problem.

Don’t assume that you have to solve every problem by yourself

Work together with others at the sprint

Other mentors or sprint attendees might have the answer!

Several occasions where we couldn’t figure out answer, but another sprinter could

you don't have to know everything--just know who to ask or how to find out. reveal your ignorance. ask the participants if they know something you don't, and tell them who you'd ask.

[YesCT]

This photo shows people at Sydney doing a lot of pointing, thinking and talking to each other. A lot of time is spent at sprints communicating; it is time well spent.

When you mentor, expose your reasoning
Ask participants questions
Reveal your reasoning when you give answers.
They think you are magically brilliant... :)
Show someone how to look up the answer you give them (api.drupal.org, d.o handbook, contributor task docs, other issues, etc.)
It's not magic, it's knowing where to find the answers.
This builds their confidence that they will be able to do this in the future without you right there.

Match task to participant

[YesCT]

We need to get permission for this photo & a higher-res version

This photo shows a sprint participant smiling. A task matched well to the participant will leave participants smiling.

Assess their language and communication skills--are they a good candidate for change notices, issue summaries, docs patches? Doc patches terrify some people, but others love to work on clear wording.
Are they already familiar with git? Rerolls might work well for them.
Are they a developer at their day job? Give them a small coding change to do to just get used to the core way of making a patch and interdiff.
Give participants a first task that they can succeed at. A simple task!
This will give them confidence to stay engaged and keep trying.
A simple task is the best first task, even for people with a lot of skills

Read the issue first, so to make sure it's a nice issue, not a controversial epic issue.
Could you understand the task quickly? if not, don't assign it.
Match the task to the participant and make their first task one they will succeed at.

[YesCT]

This photo from the Munich sprint shows folks laughing and interacting well together.

Be nice!
Treat people like peers and remind yourself that you were once new too.
During the sprint, ask people how things are going--some participants might be heads-down working, but others might be shy of asking questions. You approaching them gives them a chance to describe what they are stuck on.
Make sure people understand it’s okay to try a different task if the one they have isn’t working for them.

[YesCT]
Let’s review.

Your mission

- Help lead a sprint
- Tell us about it -- @drupalmentoring

[YesCT]
Your mission is to lead a sprint! On friday or at home, ... or both! :)
Let us know what you are planning, if you want help, and how it went.
We'll collect data and communicate information from the feedback to help future sprints.

[YesCT]

Zendoodles reworking (nudging) a three panel slide to reuse here that reuses images, sliced, from the three main points. Also probably take 'Remember' and use the key words we want them to remember.

Remember to use the contribution mentoring tools like the core office hours site and the contributor task documentation pages.
Prepare a task list in advance
Ask your participants to set up their dev environment in advance and have tasks on hand for people that dont have a dev env.

Resources

- #drupal-contribute
- @drupalmentoring
- http://drupal.org/new-contributors
- http://core.drupalofficehours.org/portland
- http://drupalladder.org
- http://drupalize.me
- http://xjm.drupalgardens.com
- http://comm-press.de/en/users/cathy-theys

[YesCT]
create the portland page (doh.org? d.o? call it something more generic?)
point out: follow @drupalmentoring for reminders about events this week.
Our slides and a handout are available on the portland drupalcon site for you to download so you dont have to write these down.

[YesCT]
Picture yourself at a sprint. The chair in this photo is waiting for you.

Now: questions!

Thanks

- Angie Byron
- Tim Plunkett
- Scott Reeves
- Kay VanValkenburgh
- Neil Kent
- Stephanie El-Hajj
- Megan Sanicki
- Holly Ross
- Pedro Lozano
- Kim Pepper
- Donna Benjamin
- drupalize.me
- The Nerdery
- Acquia
- ...And all of you!

[YesCT]

DRUPALCON
Portland
MAY 20-24 2013
What did you think?
Evaluate this session at:
portland2013.drupal.org/schedule.
Thank you!

[YesCT]