Evergreen

Intro to LaunchPad

May 27, 2021

>> GINA MONTI: 1 o'clock and we are going to be looking at the introduction to Launchpad so thank you Terran for the presentation this is track 2 and we would like to thank our sponsors, the Evergreen community development initiative for platform sponsorship and Mobius is sponsoring that captioning which is available for the session and I put the link in the chat.

If you have any questions for Terran put it in the chat or Q&A section and this is also recorded so it will be uploaded to the community YouTube in couple of weeks post conference. Terran, all set?

>> TERRAN MCCANNA: I am. First of all, welcome everyone. My name is Terran McCanna, the Terran program manager at Georgia public library service and I'm one of the Evergreen community bug wranglers. To let you know, I can do a little bit of code but I am not an actual developer with a capital D. But I do want to let you all know first off that you do not have to be developer to be able to contribute your expertise. Today's session is mainly aimed at people who are new to Evergreen or those of you who have been using Evergreen for a while but would like to learn more about what bugs have been reported and what wish list requests are out there and to take the next step and not just learning about that but contribute in feedback and prioritizing which issues get worked on and which get funded by development partners and to work on testing things. There are always bug patches being developed by different entities that are being submitted and those need to be tested to make sure they will solve the actual problem for everyone and not cause other problems. The more casters the better. With all of that I am sure you figured out by now that Evergreen is open source. Meaning there is no licensing fee and the code is freely available to look out, downloaded and customized to fit means there is no massive corporation somewhere that owns the software and that is responsible for doing bug fixes and improvements in developing new features.

So the overall health and growth of the Evergreen software is utterly reliant upon people in our community, the Evergreen community, being actively involved.

With any open source software, it is not good enough to assume somebody else is going to solve a problem. We have to be responsible for making changes happen. In whatever way that we can. That also gives us the opportunity to have more influence over how we want the software to work and what features we want added in the commercial vendor software wants to make money and who doesn't really care what the end-users think as long as they are still buying our product.

Our involvement is a community might mean contribute in funding to development projects either through an individual entity paying a developer or individual who works for an institution during development and contributing back to the community, or it could be through a joint effort such as the Evergreen community development initiative. PINES is one of the ECDI partners and we've been able to contribute funding to that in order to contribute to much, much larger development projects than we would have been able to pay for on our own.

If you're at an institution not able to contribute actual development or funding for development you can also contribute in other ways such as writing documentation or even just reporting bugs and testing proposed fixes. And that is what we will focus on here today.

The reason that we like to try to encourage more people to participate is that even though Evergreen is being used all around the world in multiple countries and multiple languages and there are thousands and thousands of frontline staff that relying on the software every day to do their jobs, the number of people that actively and consistently participate directly in development testing is less than 100 people. Matt a lot of work. I did not have my normal duties I could spend all day working on this and I would love to. But I have my normal duties as well. Which takes away from my bug wrangling. But the more people that we can get involved the better. There are technical barriers to some people but I will get more into bug squashing week and how we can work around some of those barriers for testing.

As I said in the beginning, I want to emphasize again and again that you do not have to be a software developer or system or database administrator to contribute your expertise. Those are critical skills. We would not be here without the members of the community that has those skills. But it is also essential that we involve people who actually use the software on a daily basis or who are supporting the staff that use the software on a daily basis. And have to work through workarounds and trouble shoot issues.

If you do that in your role or if you are a specialist in something like circulation or acquisitions or cataloging, then your knowledge and the way you use the software and your local workflows are invaluable for testing and feedback.

It is weird to do a presentation where I can't see people's faces. So this is where I will start talking about Launchpad. Can everyone see that? I have a set of notes here -- Gina just put the link to in check. So you don't have to take notes as I go through. This should have all of the key points that I talk about in here and a few examples of things. The link to Launchpad is at the top of the notes. This is the main Launchpad page, this is the system that the Evergreen community uses to track all of the bugs and wish list requests and progress being made on bugs and discussions about how something should actually work, that one person might think is a bug in the other thinks is the future. Patches that are ready to test, basically everything to do with the development is, goes through Launchpad at some point.

Even if it is a brand-new feature being developed, that gets put into a Launchpad bug with the code link so it can be tested. The first thing -- the first thing you need to do when you go to Launchpad is register for an account if you have not done so already.

It brings you to Ubuntu one --to let that throw you off, that is just the server software running it. You can just create an account with your email and name and password. That is all you need. I already have an account so I will login.

The main overview page of Launchpad I do not really use very often. It does have a few pieces of useful information and there are links to the main Evergreen site for the software, the Evergreen wiki including a lot of specific information about the history of Evergreen through documentation and working groups etc. so it is a catch all for everything not on the homepage. And the downloads page. If you look at the downloads page, that shows the currently supported versions, 3.7 is the most recently released version and that is one in 3.6 have the most recent point releases, 3.6.3 and 3.5, the most recent is 3.5.4.

and their release notes is the main reason to come to this page because if you are running 3.6.3 and you want to know what has been released since then if you take a look at the 3.7 release notes, you can see all the new features and an overview of the bug fixes included in that.

If you look at the older release notes you will see it has everything I 3.6, everything for the 3.6 release which has the major features, that is at the bottom and then as each point release comes out which is usually bug fixes, and that is added to this document. So if you are on 3.6.1 you can look and see what was added at 3.6.2 and 3.6.3.

At basic information about what releases are out and supported is also viable here and you can drag and drop this to seek more information.

What this is showing is everything in black, that is a released version so you can see that and everything in white is the next version on the slate. So the major releases 3.5, 3.6, 3.7 contain all the major new features that are added. So for example, 3.6 included the new Curbside Pickup feature. 3.7 -- I'm drawing a blank which is release 3.7 and I was on the release team so I should know what major features were added into it, one of thing that was added into 3.7 with some new geolocation features so geolocation information is being added to that. I can look at the downloads page and see more about what is in the 3.7 rather than remembering it off the top of my head.

You can see the new features that were added here. You get the idea. As each major release comes out they include the big features and as bug fixes are done they get added into point releases and then everything is added into the master, the master is the main core code, as each bug fix is accepted into Evergreen it gets accepted into master and at certain points that is packaged into a release. So if a new bug fix gets added to the master at the next point it gets updated it will automatically go into 3.8. When it gets packaged. And it will also get back ported into the previous point releases so 3.7.1, 3.6.5. it can always be back ported because it may rely on other code but it will be back ported as it can be included in each of those.

Of play didn't confuse everyone with that. Does anybody have any questions about releases for we move on. If you do just put them in chat and I will get back to them.

Let's look at the bugs tab. You will spend most of your time here. This bugs tab has --Jeremy said some might not know what point releases are for. Point release is set bug thesis so if you are running 3.6.2 there might be bug fixes that come out and if you upgrade to 3.6.3 it will include all those bug fixes.

This shows by default all the open bugs and wish list requests. And there is a whopping 2515 of them. Let me just look at, I will sort by number so that I could see a recent one that is easy. So this is one that was just submitted. This morning. I chose this one because it is simple. This one has a title, the description, the title of the box that's a slowed share bucket, bucket by ID. This is a typo which is a valid bug.

Some of the bug requests are more complicated than this. Elizabeth submitted this bug this morning. Two hours ago. She tagged it with the buckets tag and I will talk about tags in a moment. She included a screenshot which is nice. The version number that she took the screenshot out. It is not absolutely critical in all cases but it is helpful especially if the bug sits around for a while.

Elaine Hardy came in after that, reviewed the bug and confirmed it. I will go through some of the other things that we can do here in a little bit but I want to show you the basic anatomy of a bug.

Another thing here is that the heat level, you will hear a lot of us say please go adhere to this bug. This is a number generated based on how many people say the bug affects them. And the higher the heat number, the more people it affects and that helps the developers and the funding agencies determine what to prioritize.

So if you are logged in, you will see this bug affects X number of people does it affect you can click the sign with a little hand and say it affects me analysis this bug affects you and three other people and it increases the heat number. I don't know how the numbers work I just know the numbers go up the more people say it affects them.

If Elizabeth had not put in the buckets tab and you are reviewing it you can click there and add another tag. I will get back to tags and second because we do have an authorized tag list. On your handout the second link is a link to launch the Launchpad tags. Judith says if you click on the flame it tells about the heat. Effective users add four points per user.

The official tags are here. When you type in tag it will give you a drop-down of authorized tags you can choose from. You can type in any tag but we discourage that just for tidiness purposes. So when possible you see existing tax because it helps keep things organized, most of you are librarians and so I'm sure you are familiar with using authorized forms of things. To help keep things tidy. If you select from that list you know you are selecting something that is authorized. Some of them are not particularly clear about what they mean so there's definitions here and one of the ones I'd like to point out that is not very clear is that needs test and that has an\* because it is a specialized tag. It implies it just needs to be tested but that is not what it is actually for. It is saying it needs an automated programmatic test to be written.

It rejects it if you're not on the bug Wrangler team. Of been on for quite a while site and realize that. Only bug wranglers are authorized to add to the official list of tags Prudhomme a bug Wrangler and you are welcome to contact me if you want attack added. Right before this, curbside. Here it is. I had to refresh the page. We have a number of tickets related to curbside pickup switch is a relatively new feature as of last year. And so I added a new official tag for that. It adds its own category on top of circulation.

There are diminishing returns with adding too many tags, absolutely.

One of the reasons why tags are important, if I go back to the main bugs page they built in, search is not so great, I like Launchpad for a lot of things but search is not one of them. Only searches tickets considered open so it will not, it will only search amongst the 2500 tickets and not the ones closed so nothing that is been fixed and released in a version.

It only will search the titles and initial descriptions and doesn't search the comments and additional information that gets added. had to start out with the search there and you can search by numbers so assuming tells you to look at the ticket number such and such you can search for the number and it will work. Reading the comments in the peanut gallery. So with tags, that's organized by which tickets have the most open bugs and wish lists and it doesn't differentiate between those two so you can see cataloging has the most. The reason for that is not necessarily that cataloging has the most bugs -- or Wish List requests but the cataloging group is extremely active and participates a lot so they are very good at adding their bugs and comments here. Elizabeth, catalogers are the coolest.[Laughter].

Some of this depends upon how active the people using that tickler functionality are. And some of it is also that new, 3.6, the new staff catalog was put out. Cataloging was not quite as high before that rollout.

You can click on any one of those and see what the tickets are and by default it will sort them by priority level and a lot of them will not necessarily have a prior level, if I sort opposite for example I will see undecided. They haven't been assigned a priority level and that's a bug Wrangler activity. Many are still marked -- no one's gone in and confirmed them. So let's take a look at one example.

This one, she said it's a Wish List item. This was put in this week. It affects several different people and I will say yes this affects me. This is still marked new and normally I would say yes that I agree with this but Elaine is with my organization at PINES and it's good practice not to confirm something that somebody At your own organization submitted its best to have a different organization submit it because it puts a different perspective on it, someone might say this is, that something is a bug and someone else might disagree and they might want to work the way it's already working so at least two different organization should look at it before it gets confirmed.

So I will not submit that. As a bug Wrangler I'm going to mark that as a medium. Most things get marked as medium, fixed when convenient or schedule to fix later. Something that is low might be typo that only shows up inside the code or something like that or a log or hardly anybody will see it except for the system administrator. And then high and critical, those are reserved for things that are on fire.

If I was going to work on fixing this, if I knew how, I would assign that to myself and that would allow others to know that I was working on it. You can assign things to yourself if you are testing above. For example during bug squashing week if there was a patch for this it got loaded onto a server for testing. You could assign it to yourself to let people know that you are working on testing it and then you are done assign it afterwards. And you would click and say assign me. And then click again to remove yourself.

The milestones -- let me go back and find one with a milestone. Right now my screen isn't showing the milestones. You can click -- I have the set to show a few things preview can click on the little gear icon to show what properties of the bugs you want to see. So I will turn them all on for the sake of demonstration and then when you do it yourself you can decide what you actually would like to see. So this is all of the fields turned on which makes the page a little harder to read. But it also lets you sort by any of those things as well.

Milestones, if I sort by milestone -- I can say here is for 3.6.4 so this one has milestones set. And these get set by the bug wranglers when there is an actual patch submitted and typically what it actually is ready to go and is in testing or has been ideally signed off.

This one is little more complicated than the last one we looked at. The initial bug report and this is going to be Greek to most of you I am sure but this bug report as more information about which version he included not the Evergreen vision alone but just the open search version and he included several different tax year. The pull request tag gets added by someone who submits an actual patch ready for testing. When someone does submit a patch they upload it to GitHub which I don't expect you to understand right away if you are new.

They will include the link and they will add a pull request, if you see a pull request you know there's a patch that needs testing. If you see signed off, someone has tested the patch and signed off on it.

He set the milestone for that patch and that allows the release team to see which patches are ready to be tested and targeted into a particular bug release or major release. I hope that makes sense. This was originally recorded when 3.3 was still actively supported which it no longer is. That is why it says will not fix on 3.3 and 3.4 because only 3.5 and above are actively supported at this point.

Here are the tags, some of the information I do not look at as much so I would turn those off. A few of these that I don't look at as much. That makes it easier to read. You can sort by any of the properties you are displaying. If you want to see the newest, like the newest reported when you can sort by number. There was one that was just two hours old. You can see one that was 11 years old that is still in there. You can sort by date last updated if you want to keep track of the new activity that is going on.

You can also do advanced search. The built in search is not very good so if you do the advanced search you get a lot more flexibility in your search.

You can search for things that are fix released so if I am positive that I saw a bug in here before and I can't find it, it is possible the bug was fixed and released in a version that I just have not upgraded to yet. So you can turn that on and undo those if you just want to searchtext release.

You can also do things like search by reporter or subscriber like if I think I reported above, but I cannot remember what I call that and I cannot find it, I can just click the little magnifying glass and say pick me. And it will let me search by things I've submitted.

So these are the open things that I submitted. If I want to see things that I commented on, I could do the same thing. If I remember having a discussion in a bug comment about something I can search meitnerium the same way there. And if I want to narrow that down I can use either search terms or tags.

So if I want to see just the bugs that I commented on that had to do with the OPAC jump sure there's a few that I can search that way. Which narrows it down a little bit. There's a lot of OPAC issues out there and have commented on at least half of them.

You can also do things like search -- you can use that minus your search so if I want to for example -- things that have a pull request tag but not signed off on yet, but I can search that way. So if you want to do something like search for circulation but not, but illuminate all the billing ones you can do that for example. And if you're still struggling to find things you can rely on Google. If you do, a lot of you may know this but if you do a site ":" with a domain Google will search that site and one of the advantages of this is not only that it searches pretty much everything including all the comments but it will also search both open and closed bugs they don't have to check all the different statuses to do advanced search.

You can also use the quotation marks to search for a phrase and that sort of thing. So I could do that and sometimes find what I'm looking for this way. I do this as a last resort because it is an extra step. But it's very handy if you're having trouble finding something you are positive is there.

Another thing that you can do if you find something you are really interested in -- I will just pick one at random. If I want to follow this blog and get notified every time someone makes a comment on it. You can -- I am -- I am already subscribed to everything I think. I am not seeing the right option but if you're not subscribed to everything you would see an option here to allow you to subscribe to the bug. I could save and receive all emails about this book and that every time the bug gets updated in any way you will get an email.

If you want to do something like subscribe to all of the cataloging bug comments that come through, if you are on the main bugs page you can do subscribe to bug mall here and set up a subscription and since I am a bug angler I subscribe to everything and get notified anytime the thing gets updated but most people do not want to do that. And so if I wanted to subscribe to all the cataloging bugs I could just type in whatever subscription they might want and I can choose whether I just want to get email from the buckets opened or closed or if I want to get emailed every time a comment gets added. And then I would add a filter and then I am going to do tags which is usually the most reliable and I would put in the cataloging tag.

That is another reason we want to use, stick to the authorized tag list to make it easier. You can filter by importance or status or other pieces of information. But the tags is really the most useful. You can create this and just get notified of the things you're actually interested in. That does rely on someone actually adding the cataloging tag so if someone submits a bug and they do not have the tag and you will not get notified until someone else comes along and adds it. I keep mentioning bug squashing. Which I will come back to I promise. But during bug squashing, one of the activities you work on is making sure tags are correct in adding tags that need to be added to things that do not have tags.

I talked a little bit about the anatomy of a bug ticket. But I want to talk a little bit about the workflow at also on that handout if you know I have how to subscribe to a particular tagged set of bugs. And I have the basics of how a bug report starts only to the finish. When it gets released. So basically what happens is someone identifies a problem and instead of just going here and reporting it right away you want to do Jill Jensen make sure it is actually a bug and it is not a configuration issue or just a local issue. But it is actually re-create a ball by someone else.

One way to do this is, you can ask on one of the listserv's to see if anyone else is having a problem. Or you could go to one of the community test servers and test on a clean version of the software to see how it is working there. There is a link to the community test servers page. And these are servers that volunteers in the community are keeping running that just have Evergreen, clean Evergreen with a default set of data which is the Concerto data set which is mostly music related items. And a generic set of patrons. there is also a link there the Concerto logins so these are fake patrons loaded into the Concerto data set so you can test the issue you are seeing by logging in as one of these patrons or staff members, there are staff members too.

And you can test how it works under one of those logins. The settings on those get updated periodically and reset back to the master version so you can play around with changing the configuration settings and stuff. And test things in different ways. So if you can re-create it you want to make sure it has not already been reported by doing your searches like I talked about. And if you cannot find it anywhere sure that it is a bug, you can also do this for Wish List requests, same process, click report a bug, type in a title and sometimes it will give you suggestions to say okay, you have typed in these words and they look like some of these others, has already been reported, and if not even say no I need to report this new bug.

Add in all the information that you can, the version of Evergreen you are using and if you know the database version, etc., include a very clear description of how to re-create the bug if possible. If there is an error message you are seeing trying to either write it out or take a screenshot which you can attach. And then add tags if it is a wish list request, type the wish list tag and then what ever other tags are valid and you can attach the file here and then submit.

Once it is submitted, as I mentioned earlier someone can go in as an organization and confirm it they can also add comments or marked it as a duplicate if they know that it's been submitted elsewhere and you just didn't find it, that is fine. And then once they confirm it they mark it confirmed and they would change the status to confirmed. If this is you and you are looking at someone else's new bug and you can confirm it you can market as confirmed.

A developer can take a stab at fixing it so once they do that the developer will create a patch they think will work in the upload the patch to GitHub which I won't get into here but this is a behemoth let us say that. It is, it can take several long sessions to learn about Git, there are some very good online classes you could take, there are ones that I particularly liked that was $10 or something.

They upload it to GitHub and link it here and then they add a pull request tag to it. Someone else tests that and you do have to have access to a test server to test a patch. And then once it is tested they sign off on it and adding a signed off comment. If you are testing something and you have access to a text server either locally or through bug squashing we then -- I'm running out of time -- I didn't think I would talk this long. .

>> GINA MONTI: Not a problem.

>> TERRAN MCCANNA: This is why I have notes for you to take home. I know I will show you -- there is a way to sign off through GitHub if you are a GitHub user but if you are not you can just add this text into the comments and you would add the signed off tag and you would add this text with your name and your email address and you tested it. You are taking responsibility, you tested it and did due diligence.

Once things are signed off, one of the corps committers which is a small group of developers that knows the code really well, one of the core committers will look at it and double check the code and make sure there are no red flags. Once it passes muster they change the status to say fix committed and they add it to master. So the main core of Evergreen.

Periodically, the current release manager which changes which each major version of Evergreen, and the release team and the builds team will package set up and put it into a major or point release and change the statics to fixed release of that is where you would know where the bug is at that point.

What can you do now? As a new user for Launchpad you can find the tags you are interested in and you can add heat so if you find a circulation bug that you think is important then you can go in and had heat by saying yes it affects me. You can add tags, especially if you see something that does not already have tag, there are things that do not have tags and those things that have no tags are much more difficult to find food so if you do run across them add tags to them as relevant. You can also confirm new bugs. So for example, if you go to the advanced search you can just search for the ones marked new.

If you are showing tags you can quickly go through and see if there are any that do not have tags yet which is a good activity to do that contributes without having a lot of technical expertise or even access to a test server at that point.

I went to jump over to bug squashing week and there is a link on your handout -- we do bug squashing week twice a year sometimes three and they usually come before a major release. They are pretty much the same except bug squashing week is focused on fixing bugs and Feedback Fest making sure the software is put through the paces and tested out so nothing breaks. Bug squashing does happen there as well.

This is a great time for people to get involved for the first time because there is all lot of people that are actively looking at things and working together and what we will do is we will have several volunteers put together test servers that have bug fixes or the new version applied and everyone else can log into the test servers without having to do, run a test server themselves or configure it themselves or anything.

That is a great way to get involved. I guess I am running out of time but I want to say feel free to reach out to me, there's a few other sites and things posted on their and I also want to point out that after I decided what I was going to talk about a remember that Andrea did a great presentation on this feel free to look back on this as an addition or refresher. They did a great job on that presentation. Does anybody have any questions that I missed? Jason, thank you for helping in the conversation.

>> GINA MONTI: Seems like a lot of questions were answered in the chat so thank you for participating in that.

>> TERRAN MCCANNA: Thank you Jason and Jeremy.

>> GINA MONTI: Thank you very much, Terran, for being the wisdom of Launchpad.

>> TERRAN MCCANNA: You can add comments to extinct bug so if the initial bug report did not include all of the different factors or if you have, if you do a different workflow so you are seeing a slightly different problem but it is still related to the same bug or if you want to confirm that it is important you can add those comments as well. And anyone can confirm a bug just make sure you're not from the same organization that reported the bug. That's just for community practice.

I will put my email in the chat. Feel free to contact me directly if you have a question and I know during your first tag or confirmation or anything can be intimidating so contact me if you want to make sure you are doing it right or ask me a question before you do anything feel free I am happy to help.

Thank you everybody.

>> GINA MONTI: Thank you, Terran. We have about 30 minutes break before the session start again at 2:30 PM. Catalogers organizing in track one and stay in track two if you want to go to the report interest group.