Developer Lead Quick Start Guide

*****If you are starting as a dev lead and you have not shadowed a dev lead for a minimum of 2 weeks, talk to Ben (and Rebecca) about this immediately.*****

What is a Dev Lead at Radial?

A developer lead at Radial is the bridge between the client and the rest of the team. You commit on behalf of the team what work the team will accomplish during the sprint, and then it is your job to ensure it gets completed on time. If anyone is not able to be productive on the project, it is your responsibility to find out why and do what you can to unblock them. You will communicate progress, both setbacks as well as wins, to the client. You are responsible for listening to the clients' wants and needs, and helping them determine priorities. It is your job to make sure the client has been made aware of our processes and that they continue to follow the process outlined in the Customer Onboarding Guide. For example, the client should be emailing the team and using the team Slack channels. Help your client understand that they are working with a team, not an individual.

A developer lead does not need to be the most technical person on a project. It is OK to not have all the answers and to pull in other developers to help you answer technical questions. As the dev lead, you are not the product owner. But you should remember that you are delivering to the product owner and ensure that the Definition of Done in the Pull Request Template is complete and accurate. See the Product Owner and Developer Lead roles defined here.

Interacting with the Client

Don’t be afraid of them, they are your ally

Read the Customer Onboarding Guide in its entirety so you are on the same page as the client. Proactively and clearly communicate with clients in order to stay on the same page and try to keep communication open and positive. More information on this can be found in the main Dev Lead Guide.

Sprint planning

Sprint planning meetings include, at a minimum, the dev lead and the client. The purpose of this meeting is to confirm next priorities, and what will be worked on in the coming sprint.

You are committing on behalf of the whole team

Sprint planning may be hard early on. Ask an experienced dev lead to join your sprint planning meetings and help you plan your first couple of sprints until you get the hang of it. If you overcommit your team will be responsible for staying late to deliver for the client. If you realize early on that you have over committed, you might be able to talk to the client and change the goal. As soon as you realize you might have overcommitted, talk to Ben and your team to discuss a plan.

Under promise, over deliver

Everything will always take longer than you expect. Always. So when determining your sprint goal, plan some extra time. If you actually end up with extra time then you can do more work, because it will already be groomed and prioritized. Remember, the client will never be upset about us doing more than was expected when it doesn't cost them more money. But they will be upset if we don't deliver what we told them, and the team will be upset if they have to stay late to finish something if it could have been avoided. So always build in some wiggle room. (I suggest a day or two).

Update the client on the status of the project mid-sprint

This can be an email, a phone conversation, in-person, or even just using Pivotal Tracker if they are engaged in the process. The point is, one way or another you should let the client know if we are on track to meet our goal.

Sprint Delivery

At the end of the sprint, get together with the client (preferably in person, but at least screen sharing). Show them each feature that was built during the sprint (use Pivotal Tracker to see what you should be showing them). By showing them each feature, even if they know what you did, it helps them realize that we did actual work during the sprint, and will strengthen our relationship with them. At this meeting, if they have not yet "Accepted" the stories on PT you should encourage them to accept the stories as soon as possible. For some clients you may have to Accept the stories with them at this meeting, but that is not preferred. If you cannot get the client to Accept the stories talk to Ben, the project will most likely need to be put on hold until stories can be accepted, to avoid unaccepted work becoming a problem.

Keeping your project on track

Pivotal Tracker

Grooming

- Grooming meetings are used to estimate the size of the stories and make sure all necessary details are included in the descriptions. The meetings are also used to make sure all developers on the project understand what each story is supposed to accomplish.
- The project should be groomed every week. Yes, EVERY week, even if there have been no changes. This is important because more details and unknowns will come out of the meeting practically every time. You are responsible for scheduling and holding these meetings. By grooming with your team every week it ensures that the team knows what is going on in the project and the stories have adequate details so anyone can pick it up and start working.
- You should invite everyone that is expected to work on the project that week to the meeting.
- Go through all stories from present day up to about 2-3 sprints in the future (if it is reasonable, consider grooming all scheduled work).
- If the story points are already estimated you do not need to redo that, but make sure you read through all the notes/details and update them as needed.

Checking it daily and reporting progress/status

- Every day you should check the progress on Pivotal Tracker.
    - Are the statuses of each story accurate (started, finished, delivered)?
    - Have any stories had the same status for multiple days? Why? How can you unblock it?

Reporting about deadlines and status towards deadlines

- If it appears that you are under-resourced and will struggle to meet your goals in the amount of time budgeted for your project, let management know immediately. Ideally you want to inform management of the situation before it is a problem, not after it is already a problem.
    - Once identified, keep management and the team informed each day how progress is going.
- Likewise, inform management and the team if you are ahead of schedule
- This may be hard to determine in your early days as a dev lead. Ask for help from others to determine if it is on track. This will become easier with time and practice.

Other Daily Tasks

Check error logs

- Check Rollbar/Sentry/etc, is there anything new or critical? Should it be added to Pivotal Tracker?

Check hours budget and goal progress

- Use Freckle to check the project’s hours and compare the progress against Pivotal Tracker. Does it appear that goals will be met? If you do not know how to check this on Freckle, ask someone (everyone should be able to do this).

Are team members blocked?

- Does everyone know what to work on? Do they need anything from you or the client?