Live Operations (Live-Ops:

After a game has shipped live operations are the life and death of the product. Maintaining game balance, keeping the servers online, acquiring new users and keeping the team focused on the game becomes increasingly difficult the longer the game has been out. The frustrations and fears around the increased difficult all become high stressors if the game is not hitting KPIs. There are several subsets of live operations which are important to breakdown.

Live Operations is often about reacting to the current state of the game. This could be changing game play trends, fixing bugs, improving UI/UX, future release development and changing in game content. LiveOps is pretty similar to developing a game but in many ways it is like working on two games at once. You have the game existing in its current state and you have the game you want it to become.

Maintaining Live Code: Live code maintenance is the kind of task that few people enjoy and requires a developer to look at other people’s code. Anyone who is a programmer knows that seeing code with inconsistent naming conventions, no comments and unintuitive flow understands the importance of writing clean code. This is especially true for live games where the original developer may not be on the team anymore or a developer is quickly reacting to a live issue. No one wants to struggle through someone else’s code at 2AM only to find out they were searching down a rabbit hole. As a result, code maintenance becomes a full team effort. The more buy in the developers have for consistent practices, the more consistently variables are named and the more shared knowledge the team has about the code base, the easier life in live ops land will be. Front loading (I.E. before you have users) as much of your cleanup as possible sets up the team to have an easier time running live operations, if you can fix it before you release you will be saving yourself massive headaches down the road. If you have a roadmap for the game’s ongoing development than you can make sure to spend extra time cleaning up the components of the game that will be getting changed first. Once the game is out it is imperative that people playing the game know that you will have a consistent stream of updates to keep them interested.

Prioritizing what code should be cleaned up and which bugs to fix is a constant juggling effort. It is often impossible or prohibitively difficult to show that a specific bug has a real cost associated with it. If it is easy to show that a bug is costing you users or LTV than you need to get that bug fixed ASAP. There is nothing wrong with having a programmer stop what they are currently working on and react.