**Environments**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Environment | Activity | Permanent\* | Associated Git branch | Build trigger | Build automated \*\* | Data | Obfuscated | Backed up source |
| Local Dev | Development Unit Testing  Deployment Testing | N | Feature | Developer when ready to unit test or test deployment | N | As per Integration Testing but may also include any data required to unit test. | Y | Integration Testing |
| Local Test | Feature Testing | N | Feature | Tester when ready to feature test | N | As per Integration Testing but may also include any data required to feature test. | Y | Integration Testing |
| Integration Testing | Integration Testing | Y | Development | Feature branch merged to Development branch | Y | Cut-down | Y | Initially from Live and periodically refreshed from Live.  During a build, source is a backup of Integration as of last release (representing last stable point.) |
| Systest | System Testing | N | Release | Release manager when ready for release | N | Full | N | Live |
| OAT | OAT | N | Release | Release manager when release passed System Testing | N | Full | N | Live |
| Live |  | Y | Master | Release manager when release branch merged to Master | Y | Full | N |  |

\* An environment that is not permanent can be (in theory) spun-up when required and then spun-down when activity is finished. In practice these environments may stay on permanent.

\*\* A tool like Jenkins could watch for merges into the branch and from the name of the merge automatically build. In practice this may be manually triggered.