



Nightscout process Documentation

Release 0.0.0

Nightscout contributors

January 21, 2015

1	Quality controls	1
1.1	About	1
1.2	Social networks	1
1.3	Contributing	3
1.4	Support team	4
1.5	Sponsorship	4
1.6	Design controls	5
1.7	Documentation controls	6
1.8	Identification and traceability	7
1.9	Production and process controls	7
1.10	Nonconforming product	7
1.11	Corrective and preventative actions	11
1.12	Labeling, Packaging	11
1.13	Distribution controls	12
1.14	Record keeping	12
1.15	Servicing	13
2	Operations logs	15
2.1	2014 Operations	15
2.2	2015 Operations	15
3	Quality controls	21
4	Operations	23
5	Indices and tables	25

Quality controls

1.1 About

1.1.1 Audience

Who reviews these documents?

Contributors

As a decentralized and open project, co-ordinating efforts can be tricky, especially for new members. These documents should help people get an idea of overall roadmap, and where to look in order to best contribute their resources and talents.

Support, community, others

As reference to new members, and as reminders to old members. These serve as a handbook or guide for how to care for the community. As a by-product, they also describe what criteria or expectations are expected from people aiming to help.

As a guidebook or manual, these documents may help point to other resources people can use to better inform themselves of how Nightscout works.

Regulators

While we maintain ongoing discussions with regulators at the FDA, the changes in these documents help us demonstrate compliance while minimizing effort.

1.2 Social networks

Social networks are used to co-ordinate efforts between like-minded people, to establish a culture of safety. These social networks have features that allow people to donate time, ability, and other resources to sustain an acceptable level of quality, in an otherwise decentralized environment.

1.2.1 Github

Github is a social network built around how git, versioning control software, works. The source code and development mostly happens using git. The github social media site allows developers to co-ordinate changes using git, using established ssh keys, and other secure credentials to govern who has write access to the source code.

Teams

There are teams in the github organization. These teams govern who has commit, or write access, to the documentation and source code.

The screenshot shows the Github organization page for 'nightscout'. At the top, there are navigation links for 'People 20', 'Teams 5', 'Audit log', and 'Settings'. Below this is a search bar labeled 'Find a team...'. On the right, there are buttons for 'My teams' and '+ New team'. The main content area displays five teams:

- Owners**: 9 members · 10 repositories. Includes a 'Leave' button.
- dev**: 8 members · 3 repositories. Includes a 'Join' button.
- editors**: 8 members · 3 repositories. Includes a 'Leave' button.
- pebble-dev**: 4 members · 1 repository. Includes a 'Join' button.
- Windows Phone**: 1 member · 1 repository. Includes a 'Join' button.

Each team card shows a row of member avatars and a 'Leave' or 'Join' button.

Discussion and documentation

Github allows discussions on per line number, commit, pull request, or issue basis. Combined with other use of social media, these materials allow constant iteration on design, and implementation.

One way to visualize the data that drives the github process is through the waffle.io viewer <https://waffle.io/nightscout/nightscout.github.io>

Anyone with a github account can participate, in discussion, creating or commenting on issues, or pull requests.

Issues

Issues have a labeling system that allow them to track feature requests, bug reports, as well as design discussions.

Pull requests

Pull requests allow for review, verification, testing, and discussion of source code as it is prepared for adoption.

Continuous Testing

Projects that are using github can arrange to have Travis run tests on every commit. The results of the tests are seen in pull requests.

Analysis

Some services such as <https://codeclimate.com/github/bewest/sgvdata> offer free analysis of source code for linting and common errors.

Some services such as coveralls <https://coveralls.io/r/bewest/sgvdata> offers free code coverage analysis, to visualize which parts of the code have been exercised by the tests. <https://codecov.io/github/nightscout/android-uploader?ref=dev>

Releases

Issuing git tag for the master branch at the time of release for the new version creates a release page on github's site. These can be used to track progress of releases, which are tracked both by git hash and by semantic version number.

1.2.2 Facebook

The Facebook group, cgm in the cloud <https://www.facebook.com/groups/cgminthecloud/> is the primary home where group members discuss all things related to Nightscout, including usage stories, troubleshooting, support, and news related to updates.

A smaller support group “Nightscout Support Team” <https://www.facebook.com/groups/NSsupport/> also discusses upcoming releases announced by developers and documentation updates.

1.2.3 Twitter

The Nightscout Foundation uses twitter accounts to communicate news related to Nightscout development and progress in social media. <https://twitter.com/nightscoutproj>

1.2.4 Google

The development team uses the Nightscout Beta <https://plus.google.com/communities/116324078003160040094> google plus group to distribute and communicate updates co-ordinated with releases through the Google Play store..

1.3 Contributing

1.3.1 Contacting the team

Check the github issues to find like minded people, and discuss your topic of interest.

Check the Facebook group, twitter or other social media channels.

Contact the group developers group.

1.3.2 Contributing

Submit a pull request on github; this will notify the people maintaining the code base of your proposals.

1.4 Support team

1.4.1 Team

1) From the home page of the Nightscout.info website. “There is no support or any warranty of any kind. The quality and performance of the project is with you if you choose to use it. This is a project that was created and is supported completely by volunteers.” (This is an adaptation of the original disclaimer that was part of the “Easy Setup Guide” on GitHub.)

2) On the Nightscout.info website is the do-it-yourself troubleshooting page. <http://support.nightscout.info> Users are encouraged to use this page to solve their own issues.

3) In the nightscout.info installation documents, users are urged to post their issues to the CGM in the Cloud group. Any person leaving a comment in the forums of nightscout.info, on the Nightscout facebook page, or on the Nightscout Foundation facebook page requesting assistance, is directed to the CGM in the Cloud group.

4) The “support” volunteers are loosely organized (no set scheduling, no interview process). Once identified as being interested in support and display a team-work attitude and aptitude for problem solving, specific individuals are asked to join a Facebook group in order to discuss issues they see in the CGM in the Cloud group. However, any person that is in the CGM in the Cloud group can offer support to anyone else.

5) Individuals, not only the above-mentioned “support” group, as members of the CGM in the Cloud group, scan the posts in the CGM in the Cloud group to identify people needing assistance.

6) The only method currently used that might be used for tracking is by counting those who have successfully setup and received a certificate of completion (CGM in the Cloud Wings). It is a graphic made by a member of the support group. Any person who is a member of the CGM in the Cloud group can post that graphic to anyone else.

7) Many issues are addressed and resolved through email, private messaging and other direct communication methods. There is no capacity for capturing these conversations since they are a) private and b) unknown to anyone but the user and the person providing informal assistance.

DIY Support | The Nightscout Project support.nightscout.info

1.5 Sponsorship

<http://www.nightscoutfoundation.org/>

1.5.1 Nightscout Foundation

Volunteers created the Nightscout Foundation to curate and sponsor the activities of the Nightscout project. While we anticipate most activity being decentralized in nature, the Foundation is available to communicate information to support and development channels.

The Foundation sponsors the activities as specified in the process-controls.

Note: This is for: <http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820&showFR=1&subpartNode=2>

1.6 Design controls

Design controls are applied continuously throughout the entire software development lifecycle process.

Discussion of issues relating to design are handled in discussions on pull requests marked as exploratory, experimental, or design proposals, as well as in github issues, when issues are marked as feature request or update request of some kind.

The pull request discussions in particular provide an in depth view of which design considerations were considered at which point in time, by who, and for which reasons a decision was made.

Waffle.io can be used to get an aggregate high level overview of all issues in consideration at all stages of the process.

1.6.1 Design planning

See waffle.io and github issues, pull requests.

Can provide snapshots via api.

1.6.2 Design input

See waffle.io and github issues, pull requests.

Can provide snapshots via api.

1.6.3 Design output

See waffle.io and github issues, pull requests.

Can provide snapshots via api.

1.6.4 Design review

See waffle.io and github issues, pull requests.

Can provide snapshots via api.

1.6.5 Design verification

See waffle.io and github issues, pull requests.

Can provide snapshots via api.

1.6.6 Design validation

See waffle.io and github issues, pull requests.

Can provide snapshots via api.

1.6.7 Design transfer

See waffle.io and github issues, pull requests.

Can provide snapshots via api.

1.6.8 Design changes

See waffle.io and github issues, pull requests.

Can provide snapshots via api.

1.6.9 Design history file

See waffle.io and github issues, pull requests.

Can provide snapshots via api.

Note: For <http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820&showFR=1&subpartNode=21:8.0.1>

1.7 Documentation controls

Changes to the process are record in github for process-controls. Support team edits quickstart and troubleshooter guides according to advice from engineers. These notes are co-ordinated through README's available on github as well as the Release notes for each version.

Here are the README's:

1.7.1 Changes to the system

There is a record of all changes to the system, at the following addresses, with public ssh git addresses to match. The development teams encourage pull requests and issues to propose and discuss changes.

process-controls

<https://github.com/nightscout/process-controls/commits/master> Changes to the process-controls document reflect on-going operational processes and changes in those processes that effect quality. In many cases changes in documentation reflect changes in operation that have already occurred.

android-uploader

Readme: <https://github.com/nightscout/android-uploader/>

<https://github.com/nightscout/android-uploader/commits/master>

cgm-pebble

<https://github.com/nightscout/cgm-pebble/commits/master>

cgm-remote-monitor

<https://github.com/nightscout/cgm-remote-monitor/commits/master>

chrome-uploader

<https://github.com/nightscout/chrome-uploader/commits/master>

ios-monitor

<https://github.com/nightscout/ios-monitor/commits/master>

Note: For <http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820&showFR=1&subpartNode=21:8.0.1>

1.8 Identification and traceability

All apps follow a versioning scheme. Some apps follow semantic versioning scheme. Some of those also distribute with git-hash to double confirm against releases. Releases are git-tag'd at a hash against their published version.

- <http://semver.org/>
- <https://github.com/nvie/gitflow/wiki/Command-Line-Arguments#release>
- <http://danielkummer.github.io/git-flow-cheatsheet/#release>

Note: For <http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820&showFR=1&subpartNode=21:8.0.1>

1.9 Production and process controls

Only nightscout-contributors who have an ssh key as authorized by github have commit access. Most people are expected to fork, commit to personal branches, and then submit pull requests in public for discussion, review, testing, before merging into our mainline development process. Our mainline development process is best described by git-flow, where feature development is done on branches, development being proposed for next release is shared via dev branch, and when dev branch is ready for release, a release branch is created and merged into master.

Unit tests are often run on each commit to each branch at every step of this process, as well as code reviewed by an increasingly skeptical and wider audience.

Note: For <http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820&showFR=1&subpartNode=21:8.0.1>

1.10 Nonconforming product

1.10.1 Crash reporting

Application Crash Report for Android also known as ACRA is an open source crash reporting system for Android. It provides a mechanism for the team working on the android uploader to collect and analyze crashes or unexpected

behavior in the application. It also provides an avenue for users to manually report bugs they encounter in their daily use. The user can disable this reporting in the configuration, it is enabled by default.

On average, we receive about 10-15 reports per day. Majority (~30%) of the reports are manually generated by the user.

Currently, only the core team that works on the uploader has access to Acralyzer.

ACRA introduction

[Application Crash Report for Android also known as ACRA](<https://github.com/ACRA/acra>) is an open source crash reporting system for Android. It provides a mechanism for the team working on the android uploader to collect and analyze crashes or unexpected behavior in the application. It also provides an avenue for users to manually report bugs they encounter in their daily use. The user can disable this reporting in the configuration, it is enabled by default.

On average, we receive about 10-15 reports per day. Majority (~30%) of the reports are manually generated by the user.

Currently, only the core team that works on the uploader has access to Acralyzer.

Acralyzer

ACRA can be used with a variety of frontends. We currently use the [Acralyzer](<https://github.com/ACRA/acralyzer>) frontend for reporting. You can search and sort the bug reports.

Bug reports can triggered by 3 mechanisms:

- User requested via the feedback button
- Application crash
- Unexpected events in the uploader that the uploader thinks it can recover from

Dashboards

The Acralyzer dashboard allows us to quickly see hotspots in the code. The dashboard allows us to see spikes in reports at a glance, as well as a breakdown of reports by version.

Information reported by ACRA

- Android Package/Application name
- Code Version
- Application version name
- Uptime at the time of the crash
- When the crash occurred
- When the crash was reported
- A uniquely generated ID for the device reporting the crash
- Email address - if the user provided it
- Stack track of the crash
- Timezone associated with the uploader at the time of the crash

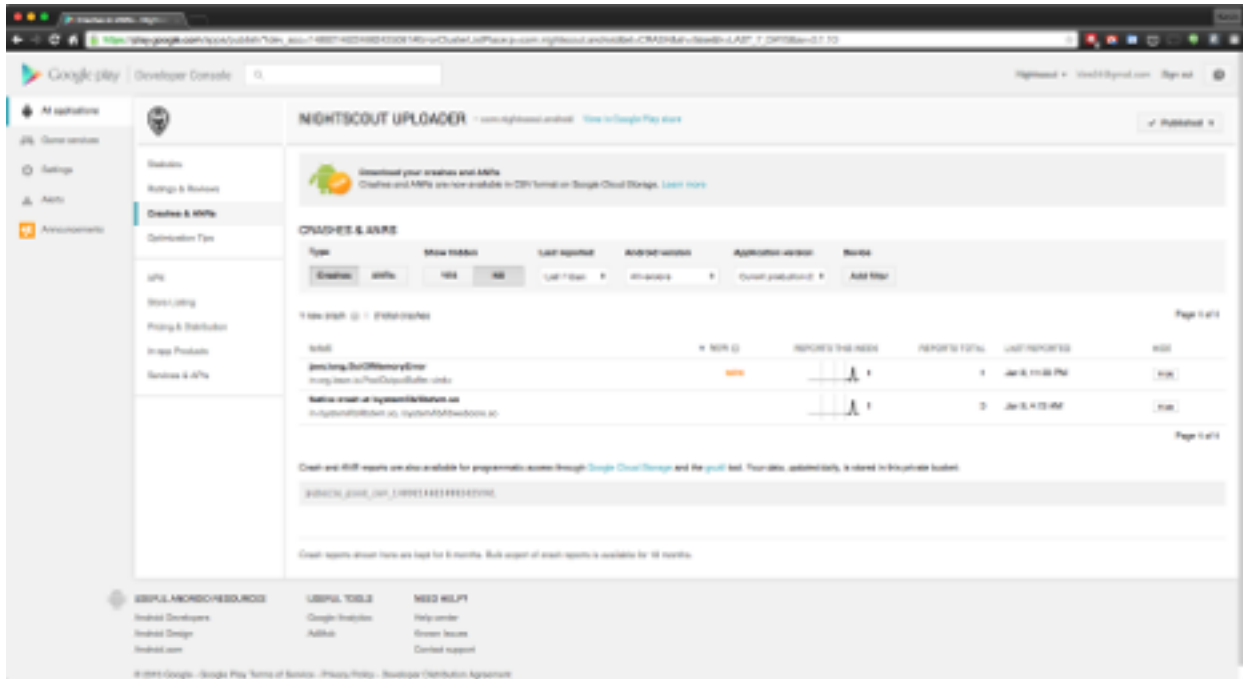
- Android version
- Available memory on the device
- Detailed hardware information related to the phone and build of android
- Configuration/State of android at the time of the crash. Example: Hidden keyboard, density dpi, orientation, etc
- Detailed hardware capabilities as reported by the device - Example: bluetooth/bluetooth_le enabled, accelerometer, cdma/gsm/wifi, camera
- Display details and state e.g. width, height, or refresh rate
- Environment details related to storage - mount points, whether or not the storage is encrypted
- Last 250 lines of the logcat - in most current versions of Android, you only have visibility to your app.
- Phone model
- Global settings - Wifi state, wifi sleep policy, backup enabled, etc
- Secure settings - Android ID, lockpattern enabled, screen saver state, etc
- System settings - Default alert sound, volume of all the different channels, hearing aid connected, haptic feedback state, etc
- All preferences associated with Nightscout. We explicitly exclude password information.
- HTTP Headers associated with the report submission as well as the client IP of the submitted report

Ongoing Future considerations of crash reporting in Nightscout ACRA's future is [uncertain](<https://plus.google.com/118444843928759726538/posts/B5BGjieGVUA>). Alternatives like [Splunk's MintExpress](<https://mint.splunk.com/>) or [HockeyApp](<http://hockeyapp.net/features/>) offer analogous services that meet or exceed our current needs.

Google play console

Applications obtained via Google play have a built in mechanism to report "Crash and Application Not Responding (or ANRs)" back to the development team. Data obtained via Google play console is a small subset of the data offered by ACRA. The play console offers statistics around how many users are affected, what devices are affected, and the versions of android affected. Beyond that it really only offers a stack track and user comments.

In the last 6 months, we have only had 4 reports via the play console. 3 of the 4 reports can be attributed to the chart and only affected android 4.3 devices. The other reported crash was due to an out of memory condition with the mongo driver.



1.10.2 2015-01

At the time of writing this document, there are roughly 1300 installations originating from the Google Play Store.

Note: <https://gist.github.com/ktind/eb2af782753f5d14fdcc>

See git flow “hotfix.”

<https://github.com/nvie/gitflow/wiki/Command-Line-Arguments#hotfix>

Note: For <http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820&showFR=1&subpartNode=21:8.0.1>

1.11 Corrective and preventative actions

Github issues are used to record github issues in public. When issues cannot be protected by patches or confirmed by unit tests, documentation is often updated, and the issues serve as public reminder of why preventative or corrective decisions were made.

1.11.1 android-uploader

<https://github.com/nightscout/android-uploader/issues>

1.11.2 cgm-pebble

<https://github.com/nightscout/cgm-pebble/issues>

1.11.3 cgm-remote-monitor

<https://github.com/nightscout/cgm-remote-monitor/issues>

1.11.4 chrome-uploader

<https://github.com/nightscout/chrome-uploader/issues>

1.11.5 ios-monitor

<https://github.com/nightscout/ios-monitor/issues>

Note: For <http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820&showFR=1&subpartNode=21:8.0.1>

1.12 Labeling, Packaging

Each app has a way to view information about the version of the software involved.

1.12.1 android-uploader

1.12.2 cgm-pebble

1.12.3 cgm-remote-monitor

1.12.4 chrome-uploader

1.12.5 ios-monitor

Note: For <http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820&showFR=1&subpartNode=21:8.0.1>

1.13 Distribution controls

1.13.1 android-uploader

<https://github.com/nightscout/android-uploader/releases>

See also, google play store releases, alpha, and beta channels. These are co-ordinated through discussion on the google Nightscout beta group.

1.13.2 cgm-pebble

<https://github.com/nightscout/cgm-pebble/issues>

1.13.3 cgm-remote-monitor

<https://github.com/nightscout/cgm-remote-monitor/releases>

Upgradeable through pull request, made easier with update-fork tool.

1.13.4 chrome-uploader

<https://github.com/nightscout/chrome-uploader/releases> The “master” branch is kept up to date with releases to the chrome app store.

1.13.5 ios-monitor

<https://github.com/nightscout/ios-monitor/releases>

Releases managed by Apple Inc.

Note: For <http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820&showFR=1&subpartNode=21:8.0.1>

1.14 Record keeping

A high level overview of the entire process is described in process-controls documentation. The documentation itself is intended to reflect how the processes in use work together ensure a culture of safety in pursuit of quality.

1.14.1 Device master record

This documents, the changes to this document, the accumulated github issues, pull request, and changelogs constitute the master design record.

1.14.2 Device history record

See the release processes for detailed historical archive on the history of releases.

1.14.3 Quality system record

See the process-controls manual, its changelog and release history, as well as the ammended operations logs.

1.14.4 Complaint files

See how support works on social media, with use case studies in the operational logs. The github issues serve as the best formal record of how issues complaints get reported, and then fixed.

Note: For <http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820&showFR=1&subpartNode=21:8.0.1>

1.15 Servicing

Note: For <http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820&showFR=1&subpartNode=21:8.0.1>

Operations logs

2.1 2014 Operations

2.1.1 Nightscout created

2.1.2 FDA and Nightscout

2.2 2015 Operations

2.2.1 Overview

In October, 2014, Nightscout contributors met with FDA to discuss Nightscout, aka “CGM in the Cloud.” This meeting is to follow up on questions remaining open for discussions during the last meeting.

Those questions centered around describing a variety of processes, known as controls, and discussed in <http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820>

- Design controls
- Document controls
- Change controls
- Testing, support
- Distribution controls
- Traceability
- Nonconformance

We’ve created a document to track each of these controls separately. What follows is a report on how those controls have played out in the past months.

2.2.2 Process description

- <https://github.com/nightscout/process-controls>

How quickly are changes or problems addressed

In some cases very quickly, in some cases weeks, months, years may be required to solve a problem.

In the instance where Dexcom released new firmware, we were able to release a new patch after 3 days. In other instances, such as adding other network protocols, or adding devices, it has taken months of considerable co-ordinated effort.

How is information on updates distributed

- Facebook
- Twitter
- Play store
- ios app store?

How are updates tracked

While development and release tracking all happens on github, updates, usage analytics, and deployments are tracked in varying ways.

cgm-remote-monitor

cgm-remote-monitor usage can be polled by analyzing the “distance” from a fork’s master to the original source’s master branch. When the distance is zero, the branch is up to date. The <http://nightscout.github.io/pages/update-fork/> tool suggests updates automatically. It’s also possible to simply delete the forked repo, and refork the original code base again, which is what support has historically recommended.

cgm-pebble

Pebble watchfaces are automatically updated by pebble. On Android this happens through the play store, and on Apple it happens when Apple approves updates.

android-uploader

The github releases page <https://github.com/nightscout/android-uploader/releases> has a list of all releases and release candidates. The releases are signed with our developer signing key, and also posted through the Google Play store for updates.

ios-monitor

Awaiting approval in from Apple.

chrome-uploader

Distributed as Google Chrome app.

2.2.3 Process

General process

Social media outlet such as Facebook allows users to post comments. Github is concentration point for all development efforts. Everything distributed through master branch is voted by multiple engineers to be ready for “production.”

As release branches are prepared, developers let people who often help people know to test new releases. Developers also use the beta and alpha channels in the Google Play store. Once we’ve done some testing and fixed the most glaring issues, the developers close the release branch, and merges to master.

We’ve also created a process-controls document to describe how social networks play a factor in co-ordinating the work of many people to produce a system that pursues quality.

Projects

cgm-remote-monitor

cgm-pebble

android-uploader

ios-monitor

chrome-uploader

2.2.4 Ongoing development

Common problems

It’s always the strings

A common problem is correctly co-ordinating the database configuration details between the android uploader and the web app. The problem is by far the most common source of use error. This particular problems probably dwarfs all the other problems.

Solution: use QR code. A QR code generator, <http://nightscout.github.io/pages/configure/> allows users to type the configuration, verify the syntax, and then copy and paste the configuration settings into Azure. This also allows the android uploader to automatically configure itself by interpreting a picture of the scanned QR code in the Nightscout app.

This particular feature was discussed on variety of social media, <https://github.com/nightscout/android-uploader/pull/93> and then integrated through pull request.

Ongoing concerns

Data privacy and security

Most often expressed as feature request:

- User would like to allow school administration access during school hours only.

Certain features we are adding require good authentication, and there are lots of legitimate requests for being able to schedule black out zones during times of the day. We've added SSL protection to help people stay secure, and have implemented alternative transports to help keep certain credentials safer. There are a few attempts at adding authentication, but hesitant to quickly bolt something on and call it secure. Support volunteers say this rarely comes up.

How to build a better dosing machine

A frequent request is for increased, step-wise integration with DIYs or similar technologies.

More devices

Medtronic support. There is a request for Medtronic support every 2 days.

Accurate reporting

Adding dialogue to test acceptance/readiness to "donate data." Need way to submit "events" to FDA, can work on tool to amend/validate qualified submissions.

2.2.5 Nexus timing change

The mobile ecosystem provides a heterogenous environment. Occasionally updates to the behavior with new models requires changing source code. In this example, Acra crash analytics were used to detect a change in managing timing of the polling behavior. The issue reported on github as issue 123 <https://github.com/nightscout/android-uploader/issues/123> used acra info to find the required information needed to solve the bug.

The information was used to test patches, which are still under review, preparing for a release, pull request <https://github.com/nightscout/android-uploader/pull/127> 127.

In this case, the issue has 16 comments over a 13 day period. These interactions include confirming the issue, finding and reproducing tests, and confirming fixes while moving it to the next stage in the process. After being merged into the dev branch of android-uploader, a release branch will include the patch into master, along with updates to the Google play store.

The screenshot shows a GitHub issue page for the repository 'nightscout/android-uploader'. The issue title is 'Nexus 9 (Lollipop) upload / connectivity issues #123'. It has 35 watchers, 24 stars, and 1 fork. The issue is open and was created 13 days ago by user 'cjbeck'. There are 16 comments. The comments are as follows:

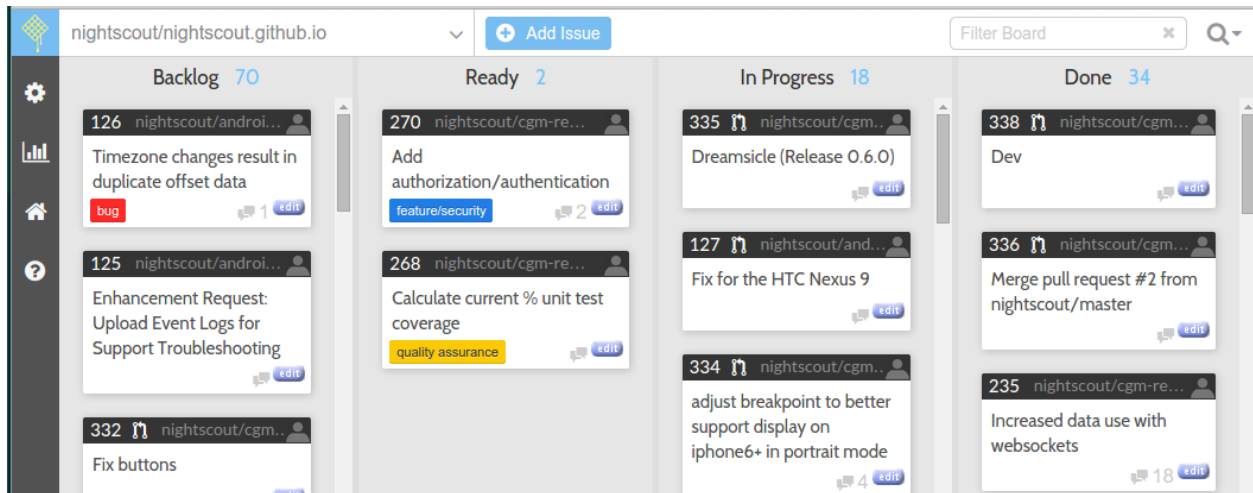
- cjbeck** (commented 13 days ago): "I'm having issues getting the Deacom update on a regular basis with my Nexus 9 (non-rooted). My previous Nexus 7 running Lollipop worked perfectly fine, but for some reason the Nexus 9 is not working as well. Are there any special settings I have to enable to get it to stay connected to the Deacom?"
- Mind** (commented 12 days ago, Collaborator): "Which version are you running?"
- cjbeck** (commented 12 days ago): "I used both the version in the play store, and tried a custom compiled one from the 1.1.11-b2 branch. Both had the same issue. I suspect it has something to do with the Nexus 9 specifically, since it appears to do something different with OTG vs the Nexus 7 (as seen in this video: https://www.youtube.com/watch?v=_sbdMuMaVzQ). It's hard to find specific information between the two though."
- cjbeck** (commented 9 days ago): "Is it possible that the 'system suspend' is the issue? Maybe the android-updater needs to do something in order to keep the tablet from sleeping in order to pull the values from the Deacom? <https://source.android.com/devices/tech/power.html#component-power>"
- Mind** (commented 9 days ago, Collaborator): (comment partially obscured)

On the right side of the issue page, there are sections for 'Labels' (None yet), 'Milestones' (No milestones), 'Assignees' (No one—assign yourself), 'Notifications' (Unsubscribe), '3 participants', and a 'Lock issue' button.

2.2.6 Case studies

These resources help illuminate how the process has played out over time.

- <https://github.com/nightscout/android-uploader/issues/123> - used acra info to solve bug
 - PR <https://github.com/nightscout/android-uploader/pull/127>
- <https://github.com/nightscout/android-uploader/pull/93> for bar code/qr code
- Dreamsicle release: <https://github.com/nightscout/cgm-remote-monitor/pull/335>
- design controls even during review: <https://github.com/nightscout/cgm-remote-monitor/pull/329>
- updating documentation <https://github.com/nightscout/cgm-remote-monitor/pull/331>
- high level overview: <https://waffle.io/nightscout/nightscout.github.io>



2.2.7 Feedback

Quality controls

Initially modeled after <http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820>, the quality controls outline a few areas, such as use of social media, and open source methodologies that are unique to Nightscout project, and then a comparison of how they compare to CFR: 21, 820.

[CITE: 21CFR820] The purpose is to describe an idealized description of of the processes that make up Nightscout, as an open source and crowd-sourced project. For ease of communication, we present several resources describing Nightscout's use of social media, and how those contribute to quality, as well as documents comparing these methods to FDA's CFR 21, part 820, outlining quality controls.

See **:cite:'21CFR820'**

Operations

Operations describes in more precise detail how the process has had an effect, referencing actual pull requests, issues, and examples from social media. Above all, this is intended to be an accurate description of Nightscout.

Indices and tables

- *genindex*
- *modindex*
- *search*