**Bug and Risk Prioritization and Resolution Process v1.2**

(Last updated April 25th, 2023)

# Purpose

The purpose of this document is to provide clear guidance on bug resolution and risk prevention. This should be used in conjunction with our [Planning and Execution](https://docs.google.com/document/d/1TSQS1EXLswR2akklOnCQEb0bfn1qrU5AeGeFv9YQm1E/edit#heading=h.l8986nhuu5fq)framework which provides further instructions on how to enter Bugs into Shortcut. This Process does not include Security Engineering concerns (e.g., security incidents, app vulnerability reports), which are addressed using the [Security Engineering SLA Guide](https://docs.google.com/document/d/1I2ll_Z4MDoKcopEpHi5T2GeDrtGLOLNqR4ZamwC9CUM/edit#).

It follows these principles:

* a good process will have different people with the same information get to the same decisions most of the times a member is affected by the bug
* **Class**:­ the type of product requirement it causes to fail
* **Risk**: likelihood the risk of a bug’s impact materializing within 3 months

Output

* **Priority**: urgency to fix the bug or vulnerability

# Summary

| **Priority = ¼ \* impact + ¼ \* members affected + ½ \* class + risk**  Rounded up or down by Product, with max value of 4 |
| --- |

\*\*Significant changes between versions are highlighted in yellow\*\*

| [**Impact**](#kix.eeyiumiwbjwr) | **Effect of the bug on a single user or Episource reputation** |
| --- | --- |
| 0 | Affects [Critical Data](#kix.u9um9esleirb) or critical and time-sensitive functionality in [Major Feature](#kix.btzmky6g6bap) without a workaround |
| 1 | Affects [Critical Data](#kix.u9um9esleirb) or critical and time-sensitive functionality in [Major Feature](#kix.btzmky6g6bap) with a difficult workaround |
| 2 | Affects non-critical data or non-critical functionality in [Major Feature](#kix.btzmky6g6bap) or critical and time-sensitive functionality in other features |
| 3 | Affects temporary or peripheral data or functionality |
| 4 | Affects aesthetics, professional look-and-feel |

| **Members/Clients Affected** | **The number of members, member data, or clients affected** |
| --- | --- |
| 0 | Affecting all or almost all members, member data, or clients (>95%) |
| 1 | Affecting more than 2/3 of the members, member data, or clients (>67% and <=95%) |
| 2 | Affecting about half the members, member data, or clients (>33% and <=67%) |
| 3 | Affecting about 1/3 of the members, member data, or clients (>5% and <=33%) |
| 4 | Affecting no or very few members, member data, or clients (<=5% of members/member data) |

| [**Class**](#kix.4sivithqj75l) | **The type of product requirement it causes to fail** |
| --- | --- |
| 0 | Critical system stability; Critical system reliability; Critical system availability; Security (see separate [Security Engineering SLA Guide](https://docs.google.com/document/d/1I2ll_Z4MDoKcopEpHi5T2GeDrtGLOLNqR4ZamwC9CUM/edit#)); Financial liability; Data storage; Compliance |
| 1 | Performance; Efficiency; Scalability; Copyright; Access Control; |
| 2 | Functionality; Logic or Calculation; Compatibility; Interoperability; Accuracy; Testability; |
| 3 | Usability; Learnability; Readability; Documentation; Consistency; Workflow (“feel”) |
| 4 | Typographical or grammatical; Aesthetics; Appearance or Cosmetic (“look”) |

| [**Risk**](#kix.p9lrx69duatv) | **Likelihood a bug will manifest the above Impact within 3 months** |
| --- | --- |
| 0 | Impact has already happened |
| 1 | Strong certainty the impact will happen within 3 months (>95%) |
| 2 | Possible chance the impact will happen within 3 months (>50%) |
| 3 | Rare chance the impact will happen within 3 months (>10%) |

# Prioritization & Service Level Agreement

The SLAs have been [defined](https://docs.google.com/document/d/1e1pEmcbu5-rmMDZml0gLSmDkU-S_2cAV61AY7iEQxak/edit#) with our customers' interest in mind, and how we affect their business and workflows every day, while also minimizing unnecessary context switching to our teams.

When we have concerns that SLAs are too strict or too loose, we review them, in doubt leaning towards our customers.

| **Issue Priority** | **On-call activity** | **Triage Response Time** | **Resolution Time Target** | **Update Frequency** | **Release** |
| --- | --- | --- | --- | --- | --- |
| 0 | Within response time SLA, triage, and start fixing. Work time includes weekends and non-business hours. | 20 minutes  Announce bug in [#p0p1-bugs](https://episource.slack.com/archives/C03FFSFKTEK) Slack channel and other communication channels as necessary | 1 hour  Advise [#p0p1-bugs](https://episource.slack.com/archives/C03FFSFKTEK) and other communication channels of resolution as soon as possible.  Conduct and publish root-cause analysis to [#p0p1-bugs](https://episource.slack.com/archives/C03FFSFKTEK) within one week of resolution | Hourly first 24 hours then minimum 2x per day  Post in [#p0p1-bugs](https://episource.slack.com/archives/C03FFSFKTEK) and other communication channels as necessary | Hotfix Process |
| 1 | Within response time SLA, triage and start fixing. Work time includes weekends and non-business hours. | 1 hour  Announce bug in [#p0p1-bugs](https://episource.slack.com/archives/C03FFSFKTEK) Slack channel and other communication channels as necessary | 6 hours  Advise [#p0p1-bugs](https://episource.slack.com/archives/C03FFSFKTEK) and other communication channels of resolution as soon as possible.  Conduct and publish root-cause analysis to [#p0p1-bugs](https://episource.slack.com/archives/C03FFSFKTEK) within one week of resolution | Hourly first 24 hours then minimum 2x per day  Post in [#p0p1-bugs](https://episource.slack.com/archives/C03FFSFKTEK) and other communication channels as necessary | Hotfix Process |
| 2 | Triage and if possible, implement a quick fix, if not set up meeting with product to prioritize within current sprint while meeting resolution time SLA. Work should only be done during business hours. | 1 business day | Up to 5 business days | Daily | Scheduled Release or Hotfix Process |
| 3 | Within response time SLA, triage and if possible, implement a quick fix or else add to the top of the appropriate team’s backlog | 1 business day | 3 weeks | As needed | Scheduled Release |
| 4 | Product will triage. | N/A | 90 days (recommended but not currently enforced) | N/A | Scheduled release |

# Definitions

| **Term** | **Description** |
| --- | --- |
| Bug | A bug is when the product does not operate as intended in production. |
| Critical Data | Critical Data is PHI, PII, or data required to perform the function of a Major Feature |
| Major Features | Coding   * Chart Allocation * Coding UI * NLP UI * L2 / MA Automation * Provider Search * Claims Data Validation * Output File Generator * Postgres DB - DB Sync * Async Output * Chart Mover * epiCoder Dashboards   epiAnalyst   * System availability * Downloads * Dashboards * Chase lists(Create/download existing) * Member/Provider index and filters * HCC registry * Campaigns   epiEncounter   * Login Page * Import File - Inbound/Response * Encounters - Search/Edit * Submit File - Outbound   epiHub   * Carequality engine * Swagger UI * epiHub primary database * Delivery of documents (Rabbit MQ & PDF Consolidator)   HRA   * Submitting EHR Assessments * Syncing EHR Assessments * Launching epi.clinic for telehealth visits * Outbound & Inbound Calling * Sending Devices * Scheduling * Loading/accessing member profiles * Self-Service Tools * Client Deliveries * Claims Processing   MRR   * Chase Lists * Member Profiles to User Profiles * Packet Templates * Reports (Operational or Financial) * Bifrost File Processing * Bifrost Fuzzy Search * Kraken Services * Outbound & Inbound Calls * Scheduling * Provider Packets * Provider Portal * ScanTech Appointments   ACO Solution (fka epiDW)   * Log in/out * Viewing canned reports * Exporting data into csv or equivalent format |
| On-Call Engineer | An engineer who has worked at Episource for more than three months.  One engineer from each engineering team (Product Eng & Bus Sys, Data Acquisition & Native Apps, Web Apps, Data & Comm Services, Data Access) |
| On-Call Engineering Manager | One Engineering Manager, Engineering Director, Engineering VP |
| Manager of Engineers | The person who manages other engineers including the On-Call Engineer. This is not related to the title but the role a person may fulfill. |
| Engineering Group Leaders | All engineering management (Manager, Director, VP, CTO) |
| Shipping Lead | A project team’s lead is responsible for prioritizing work, but not for deciding on technical solutions. |
| Rotation and Escalation System | System used for escalations and on call rotations |

## 

## Impact

Impact rates the effect of the bug on a single user or the reputation of Episource when it is encountered. It indicates how much the bug will interfere with the user's ability to perform a task, or do their job when it occurs, as well as how difficult it is to find and use a workaround (if there is one). The table below gives a description of each impact level. Note that the term “functionality” in this table is defined loosely, and can refer to any action or request made by the user, or any operation or function that the product performs. The item in square brackets is a possible emotional response of the user.

When rating the impact of the bug, do not consider the total number of users of the product that may be affected by the bug. Nor consider the relative importance of the feature where the bug is found. For impact, only consider how much the bug interferes with a user completing a task or doing their job when it occurs.

| **Impact** | **Description** |
| --- | --- |
| 0 | * Affects critical data or functionality in major feature that severely affects users or could have irreparable harm to Episource reputation due to other causes, with no workaround (e.g., user unable to complete a critical task necessary to perform his/her job) * Huge impact on productivity or efficiency. Most would say that the feature is unusable. * Effect: Causes much anger and pain, and some crying; significant revenue / brand equity loss |
| 1 | * Affects critical data or functionality in a major feature or could also pose some reputation harm to Episource, with a difficult workaround (e.g., user is able to complete a critical task with some difficulty, using a workaround that is not obvious and/or is hard to use). For data loss or corruption, the problem is difficult to detect or correct by the user. * Major impact on productivity or efficiency. Many would complain the feature is unusable. * Effect: Causes anger, dismay, and some swearing; significant revenue / brand equity loss |
| 2 | * Affects non critical data or functionality in a major feature, or can only be found internally and could still pose harm to the reputation of Episource, with an easy workaround. For data loss or corruption, the problem is temporary, and/or is easily detected and corrected by the user. * Moderate impact on productivity and efficiency. Some would say the feature is hard to use. * Effect: Causes frustration, annoyance, and some muttering under breath |
| 3 | * Affects temporary or peripheral data or functionality. Low impact on productivity and efficiency. Few would say that feature is hard to use. * Effect: Causes eye rolling and scowls. Will annoy nitpickers, no material revenue impact |
| 4 | * Affects aesthetics, professional look and feel, or a sense of “fun” * No impact on productivity or efficiency, except maybe for power­ users * Effect:May make you laugh instead of cry, if you even notice, no material revenue impact |

# 

## Members/Clients Affected

Rates how likely a member, member data, or client (depending on the value stream) is affected by the bug. In many ways, it is related to how close the bug is to the main path (or critical path, or core functionality, or key operations) of the feature. It also can be thought of as the probability that a typical member, member data, or client will be affected by the bug. Thus, the members, member data, or clients affected by an intermittent bug (i.e., a bug that has a dependency on time, random or unknown factors) should be downgraded relative to how infrequently it occurs.

Please note that the Members/Clients Affected rating is relative to the feature area, not the overall product. Thus, a bug in a rarely used feature can still have a high priority if it affects every member, all member data, or every client. Conversely, a bug in a very commonly used feature can have a low priority if it doesn't affect many members or member data.

| **Members/Clients Affected** | **Description** |
| --- | --- |
| 0 | Affecting all or almost all members, member data, or clients (>95%) |
| 1 | Affecting more than 2/3 of the members, member data, or clients (>67% and <=95%) |
| 2 | Affecting about half the members, member data, or clients (>33% and <=67%) |
| 3 | Affecting about 1/3 of the members, member data, or clients (>5% and <=33%) |
| 4 | Affecting no or very few members, member data, or clients (<=5% of members/member data) |

## Class

A bug's class is determined by the type of product requirement it causes to fail, and the ranking or relative importance of that requirement to the project goals and priorities. The type of requirements assigned to each *class* is set by the program or product manager. The table below gives an example of how a typical software project might classify bugs according to the type of failed requirement. For this project, stability, reliability, and availability bugs are given a higher classification (i.e., closer to zero) than, say, performance or usability. Your project priorities may differ.

| **Class** | **Description** |
| --- | --- |
| 0 | Stability; Reliability; Availability; Testability; Security (see separate [Security Engineering SLA Guide](https://docs.google.com/document/d/1I2ll_Z4MDoKcopEpHi5T2GeDrtGLOLNqR4ZamwC9CUM/edit#)); Liability; Storage (data loss/corruption); Accuracy (billing, accrual, payments), Certification (blocker) |
| 1 | Performance; Efficiency (use of resources: memory, disk, CPU); Scalability; Copyright |
| 2 | Functionality; Logic or Calculation; Access Control; Compatibility; Interoperability; Workflow ("use") |
| 3 | Usability ; Learnability; Readability; Documentation; Consistency; Workflow (“feel”) |
| 4 | Typographical or grammatical; Aesthetics; Appearance or Cosmetic (“look”) |

## Risk

Risk rates how likely it is that the impact of a bug will materialize in the next three months. This allows us to address bugs before they occur and avoid the manifestation of an impact. Estimating the chance of a future event is difficult and unverifiable. Unlikely events are not managed through this process. They can be added to the backlog through regular backlog management.

An example for clarity. Suppose a system has 100GB of storage available to it. If the storage is exhausted before it is expanded, there will be a significant impact on a large number of users. Until then, the system will operate normally.

* If the storage has been exhausted and users are affected, then the “impact has already materialized” and the Risk score for this bug is 0. This could well be a P0 bug.
* If there is 10GB remaining and we are sure to be adding 1GB of data per week, there is a “strong certainty the impact will materialize within 3 months (>95% chance)” and thus the Risk score for this bug is 1.

In both of these cases, a bug exists in the sense that there is risk of an impact materializing *if corrective action is not taken*. (It’s a fair argument that running out of storage space isn’t strictly a “bug” per se and we would welcome a better example.)

| **Risk** | **Description** |
| --- | --- |
| 0 | Impact has already materialized |
| 1 | Strong certainty the impact will materialize within 3 months (>95% chance) |
| 2 | Possible chance the impact will materialize within 3 months (>50% chance) |
| 3 | Rare chance the impact will materialize within 3 months (>10% chance) |

## Special Handling of P0 and P1 Bugs

Due to the nature of P0 and P1 bugs, we require teams to respond immediately as follows:

* **When the bug is reported:** Announce the bug as soon as possible after it is reported to the [#p0p1-bugs](https://episource.slack.com/archives/C03FFSFKTEK) in Slack with the Shortcut Story ID, a link to the bug in Shortcut, and a description of how the priority was calculated. Continue to announce stakeholders via existing channels as well (e.g., the stakeholder may be one or more external customers without access to our Slack instance). The first person who becomes aware of the bug is responsible for reporting it in Slack, the Engineering Lead is responsible for ensuring that this happens within the SLAs.
* **While the bug is being worked on:** Provide hourly updates to the [#p0p1-bugs](https://episource.slack.com/archives/C03FFSFKTEK) Slack channel for the first 24 hours, then minimum 2x per day after that. Continue to provide updates to stakeholders via other channels as well, if necessary. The Engineering Lead can delegate this update responsibility to the Product Owner or Builders on the team with their consent, but remains accountable for ensuring that updates happen within the SLA.
* **When the bug is resolved:** Announce the resolution in the [#p0p1-bugs](https://episource.slack.com/archives/C03FFSFKTEK) Slack channel as soon as possible after resolving the issue. Announce resolution to stakeholders via other channels as well, if necessary. The Engineering Lead can delegate this update responsibility to the Product Owner or Builders on the team with their consent, but they remain accountable for ensuring that the issue is resolved and the resolution announcement occurs within the SLA.
* **After the bug is resolved:** Complete an [After Action Review](https://docs.google.com/document/u/1/d/1RtxqTVYvVJaALO9saBKEgd6UkMX86Prl4d1zZGHP_v8/edit) of the bug within one week. It is best to ensure that the AAR happens as soon as possible after the incident is resolved so that key details are more likely to be remembered and the team can take rest. The Engineering/Support Lead for the team is responsible that this process occurs, that it is timely, and that it is of high quality. The team's Product Owner is responsible for ensuring that the short- and long-term actions resulting from the AAR are prioritized correctly (usually the highest priority work for the team).

## Escalation of Unacknowledged Bugs

If a bug has not been acknowledged in the time specified it will be escalated to the next tier automatically based on the following periods.

It is expected that the Tier 2 or Tier 3 person should acknowledge all escalated bugs and not reassign them. It is the expectation of these tiers to ensure that the acknowledger sees the incident through to completion.

| **Issue Priority** | **Escalation Window** | **Tier 2** | **Tier 3** |
| --- | --- | --- | --- |
| 0 | 24x7x365 | 15 minutes | 45 minutes |
| 1 | 24x7x365 | 30 minutes | 3 hours |
| 2 | 5am - 8pm Pacific Timezone | 12 business hours | N/A45 business hours |
| 3 | 5am - 8pm Pacific Timezone | 24 business hours | N/A48 business hours |
| 4 | N/A | N/A | N/A |

# 

## Escalation of Unresolved Bugs

If an issue will not be resolved or has not been resolved by the time alloted in the matrix below then the issue should be escalated. Escalation will happen automatically for unresolved incidents.

| **Issue Priority** | **Tier 2** | **Tier 3** |
| --- | --- | --- |
| 0 | 45 minutes | 90 minutes |
| 1 | 90 minutes | 3 hours |
| 2 | 5 business days | N/A |
| 3 | 10 business days | N/A |
| 4 | N/A | N/A |

## Tier Definitions

Issues may come in from automated alerting and/or be internally escalated. P4 Issues will send notifications to Product Managers for triage based on the area label. Issues should be handled as described by the following tiers based on priority

| **Priority** | **Tier** | **Assignment** | **Patch Authority** | **Resource Authority** |
| --- | --- | --- | --- | --- |
| P0 to P1 | 1 | On Call Engineers | Minor patch updates w/o code review | Can assign to other On Call Engineers |
| 2 | On Call Engineering Managers | Multiple module or significant impact change | Can assign to anybody within Episource engineering organization including partners |
| 3 | Engineering Group Leaders | Data migration, regional disaster recovery | Can assign to anybody within Episource organization including partners |
| P2 to P3 | 1 | On Call Engineers | Minor patch updates |  |
| 2 | Manager of Engineers | Multiple module or significant impact change |  |
| P4 | 1 | Product Managers | N/A | Product team will triage and bring into an engineering team’s backlog as appropriate. |