**USE CASE: CDS** using DRAFT v2.2 Patient-Safety Risk Framework

|  |  |  |  |
| --- | --- | --- | --- |
|  | Lower risk | Medium Risk | Higher Risk / More Attention |
| Purpose of software product | Information-only; purpose is transparent and clear | Makes recommendations to user | Automated decision making (e.g., intelligent IV pump, AED) |
| Intended user(s) | Targeted user(s) are knowledgeable and can safely use product | Makes recommendations to knowledgeable user | Provides diagnosis or treatment advice directly to knowledgeable user |
| Severity of injury | Very low probability of harm | Potential for non-life threatening adverse event | Life-threatening potential |
| Likelihood of risky situation arising | Rare  (<1 per 100,000 patient-years) | Unpredictable, but risky situation arises > 1:100K pt-yrs and < once a year | Common (arises once per patient-year) |
| Transparency of software operations and data and included content providers | Software output is easy to understand and its “calculation” (data and algorithm) transparent | Software operates transparently and output is understandable by software expert | “Black box” |
| Ability to mitigate harmful conditions | Human intermediary knowledgeable and empowered to intervene to prevent harm | Human intermediary may be (but not routinely) involved | Closed loop (no human intervention) |
| Complexity of software and its maintenance | Application of mature, widely adopted technologies with information output that is easy to understand by the user | Medium complexity. Testing procedures exist that reliably assess patient-safety risk profile of product. | Complexity of data collection and “transformation” involved in producing output is significant. Difficult to test reliably for all safety risks |
| Complexity of implementation and upgrades | The “build” and configuration of the software is straight-forward and does not materially affect the integrity of the output. Safety upgrades can be accomplished easily. | The “build” and configuration of the software is moderately complex, but “guard rails” significantly limit types of changes that might induce life-threatening risk. | The “build” and configuration of the software is complex and can introduce substantial changes that can induce serious risk. Limited or no “guard rails.” |
| Complexity of training and use | The software system output is clear and easy to interpret. Minimal training needed. | Moderate complexity. Less than 2 hr of training required. | The complexity of the user interface and density of data presented can cause important errors or oversights that can lead to serious risk. Formal training necessary. |
| Use as part of more comprehensive software/hardware system | Used as a standalone product, or output is unambiguously used as part of larger integrated system. Certified to specific hardware. Redundancy reduces single points of failure | Software interacts with 1-3 other systems with mature, well described interfaces | Almost always used as part of a larger software system AND output is subject to interpretation or can be configured in multiple ways whose mis-interpretation may induce harm. [e.g., DDI thresholds]. |
| Network connectivity, standards, security | Wired or tightly controlled wireless spectrum compliant with standards | Unregulated spectrum, but low risk of interference | Wireless using unregulated spectrum; proprietary interfaces |

Exemplar 3 – Clinical Decision Support Software

**Description of Exemplar**

Software module integrated or interfaced with EHR that invokes programming logic that draws on a knowledge base to provide user feedback (e.g., alert, recommendation, reminder) when certain pre-specified conditions exist

**Target users**

* Qualified EHR users

**Implementation/configuration**

* Must properly configure interaction of:
  + Knowledge base (e.g., drug database, drug interaction database, health maintenance rules, disease management guidelines)
  + Inference engine
  + Output/display configuration
* If knowledge base not supplied, must configure CDS logic to produce appropriate notification/alert based on specific conditions

**Severity of injury**

* Life-threatening

**Likelihood of the risky situation arising**

* Inference engine defect
* Improper configuration of components
* Knowledge base error
* Inference engine programming/configuration error
* Unanticipated condition

**Transparency of software operation, data, and knowledge content sources**

* Visible to analyst, but not to user

**Ability to mitigate harmful condition**

* Professional user receives and acts upon CDS message

**Complexity of software and its maintenance**

* Requires content knowledge commensurate with CDS domain covered

**Complexity of implementation and upgrades**

* Upgrades must keep up with changes of relevant domain knowledge
* Interactions amongst CDS logic possible

**Complexity of training and use**

* Requires modest training of end user

**Use as part of more comprehensive software/hardware system**

* If external component, must ensure interoperability of input data and output recommended actions

**Network connectivity**