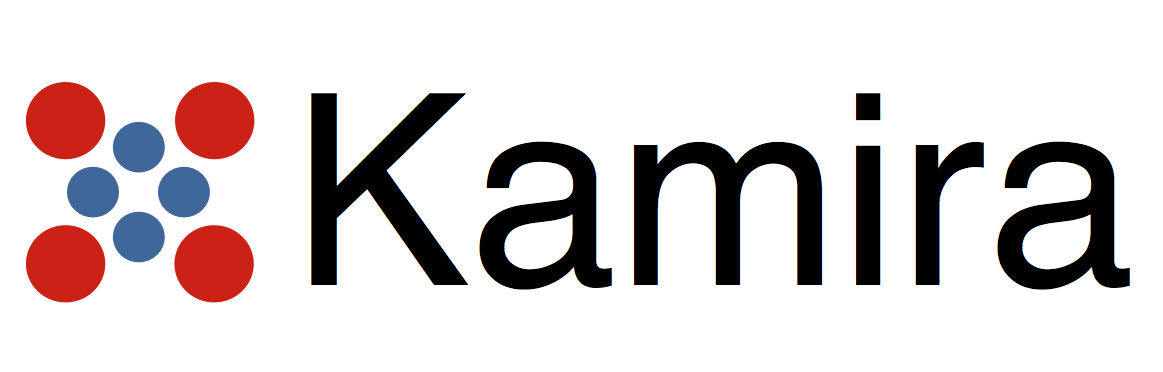
November 2012

Complexity Analysis of Meaningful Use Stage 2 Ambulatory Clinical Quality Measures

|  |
| --- |
|  |
|  |

****

**An Open Source Clinical Quality Measure Modeling Framework**



Meaningful Use is an HHS program tasked with promoting the use of electronic health records to improve health care in the United States. The Health Information Technology for Economic and Clinical Health (HITECH) Act provides the Department of Health & Human Services (HHS) with the authority to establish programs to improve health care quality, safety, and efficiency through the promotion of health IT, including Electronic Health Records (EHRs). The goal of Meaningful Use is to improve the quality of health care in the Unites States by increasing the use of EHRs. As part of the MU programs, Clinical Quality Measures (CQMs) are tools to evaluate the processes and outcomes of healthcare providers, focusing on effective, safe, efficient, patient-centered, equitable, and timely care. Considerations for linking CQMs results with Pay for Performance (P4P) programs could result in healthier outcomes for patients with a decrease in costs.

In order for CQMs to be effective metrics used in P4P programs, there should be a causal link between the CQMs and both measurable improvements in the quality of health and a reduction in the overall cost of care. To support this, CQMs need to be designed to allow for rapid implementation and testing in order to validate accuracy of CQM’s performance. The overall design of CQMs is based on the clinical experience and findings of healthcare experts. However, CQM development processes lack rigorous considerations of the complexity of the CQM algorithmic logic. In particular, high complexity in the MU Stage 2 CQM specifications will likely lead to significant effort to implement and test in EHR systems. This will lead to increased implementation times, increased likelihood of software errors, increased difficulty in testing, and an increased cost in EHR products integrating these overly complex CQMs.

MITRE believes that examining and maximizing the quality of CQMs is critical to achieving transformational performance in both quality and cost in the US healthcare system. MITRE has invested in an internally funded research project to examine how the quality of CQMs can be measured and improved, “*Kamira*”. MITRE has deep experience in designing and developing open source tools related to meaningful use and CQMs, such as the MU CQM reference implementation popHealth, and the MU Clinical Quality Measure testing and certification tool Cypress. This work has provided MITRE with a unique perspective to provide useful feedback on CQM quality to the broader CQM community. The initial focus of MITRE’s work on Kamira has been to examine the algorithmic complexity of CQMs as defined in Meaningful Use Stage 2.

MITRE's assessment shows that the MU Stage 2 Ambulatory CQMs have significant levels of complexity. This will make manual implementation and testing of the CQM logic in EHR systems more difficult. MITRE recommends examining and tracking measure complexity on an ongoing basis with the goal of clearly understanding the trade offs between complexity and utility.

Many of the MU Stage 2 CQMs specify a series of filters to focus attention on patients by demographics such as age and gender, patient-level clinical attributes such as healthcare provider encounters as well as conditions, and care provided by healthcare providers such as procedures, tests, labs, and immunizations. Clinical exclusions and exceptions to the CQM logic are also considered for individual patients by considering patient-level attributes in the exclusion and exception logic. This exclusion and exception logic provides allowable reasons for healthcare providers to not perform the quality of care specified in the CQM based on clinical considerations.

As an example we can take a superficial look at a MU Stage 1 CQM NQF0055 Diabetes Eye Exam, which calculates the percentage of patients aged 18-75 years with diabetes (type 1 or type 2) who had an eye exam within two years. The initial patient population is the set of patients who are aged 18-75. The denominator contains those patients with diabetes who have had inpatient or outpatient encounters during the measurement period. The numerator contains those patients who have received an eye exam during the measurement period. The exclusions include a number of conditions, such as a diagnosis of polycystic ovaries.

CQMs are implemented in software, and the complexity of the implementation is directly linked to the complexity of the specification. Complexity of CQM logic in software systems has several implications:

* More complex software takes longer to implement and is therefore more expensive to develop and deploy into products
* More complex software has a higher likelihood of containing undiscovered software errors and bugs
* More complex software is more difficult for implementers to maintain
* More complex software is more difficult to test and validate for correct behavior

Complexity of software can be measured in a number of different ways. The most basic is to simply examine the number of lines of code (LOC). However, this metric is very crude, and does not measure the quality of the software implementation. LOC’s primary value as a metric is volume machine instructions.

Alternative approaches such as Cyclomatic Complexity examine the number of independent paths through a program[[1]](#footnote-1). Cyclomatic Complexity is calculated by examining the control flow graph of a program, where

Cyclomatic Complexity = E - N + p

E = the number of edges of the graph

N = the number of nodes of the graph

p = the number of connected components

Studies show that Cyclomatic Complexity has a direct impact on bug frequency as well as program understandability and testability. Cyclomatic Complexity is commonly used to evaluate code against threshold values as an indicator of risk[[2]](#footnote-2):

|  |  |
| --- | --- |
| **Cyclomatic Complexity** | **Evaluation** |
| 1-10 | Simple, low risk |
| 11-20 | More complex, moderate risk |
| 21-50 | Much more complex, high risk |
| > 50 | Untestable, excessive risk |

The NIST Structured Testing methodology suggests limiting Cyclomatic Complexity of code modules to 10, with 15 being permissible in some circumstances[[3]](#footnote-3).

MITRE implemented the MU Stage 1 CQMs in JavaScript as part of the popHealth project[[4]](#footnote-4). For MU Stage 2, CQMs are specified and available using the Health Quality Measures Format (HQMF)[[5]](#footnote-5). MITRE has developed software to consume HQMF and automatically produce the appropriate JavaScript code.

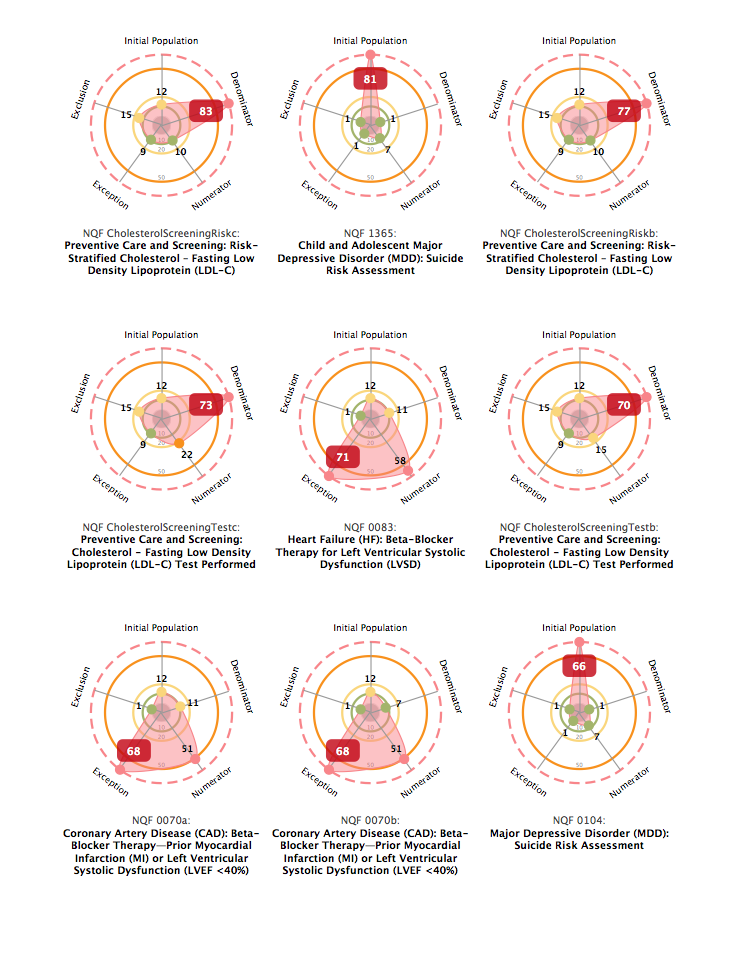
By using the open source JSMeter[[6]](#footnote-6) software package to measure complexity of the CQM JavaScript implementation of the HQMF XML, MITRE was able to measure the complexity of the CQMs in an entirely machine automated process. MITRE does want to highlight one caveat: MITRE’s findings measure MITRE’s implementation of each CQM rather than the abstract CQM logic. Because there are various ways to implement a particular specification of a CQM in software, it is possible that individual implementation choices will impact the measurements made, and either increase or decrease the CQM complexity findings relative to other software systems.

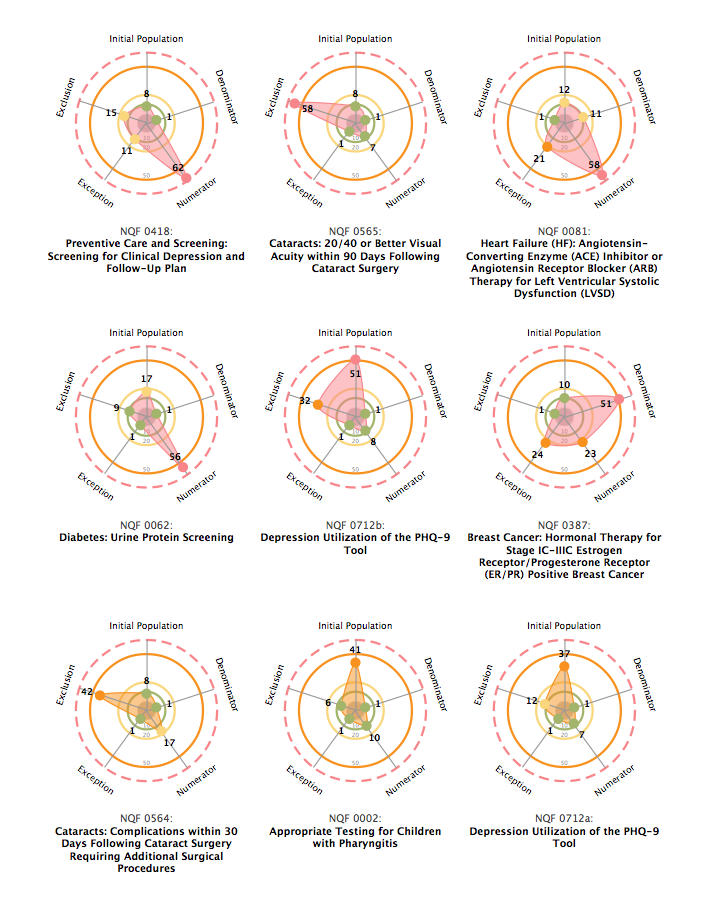
A visual representation of complexity is useful in order to quickly grasp relative complexity and note trends and outliers. One approach to the visualization of Clinical Quality Measures is a Kiviat chart. Kiviat charts, sometimes referred to as Radar Charts, are two dimensional, multi-metric illustrations that use a variable number of evenly distributed radii. Each spoke is associated with one metric. The length of a value overlaid on each radius is proportional to the magnitude of the metric relative to the maximum value of the variable across all data points. For each MU Stage 2 CQM analyzed, we measured the individual Cyclomatic Complexity of the logic for calculating the Initial Patient Population (IPP), Denominator, Numerator, Exceptions, and Exclusions components. MITRE’s findings for all of the MU Stage 2 CQMs are included at the end of this document.

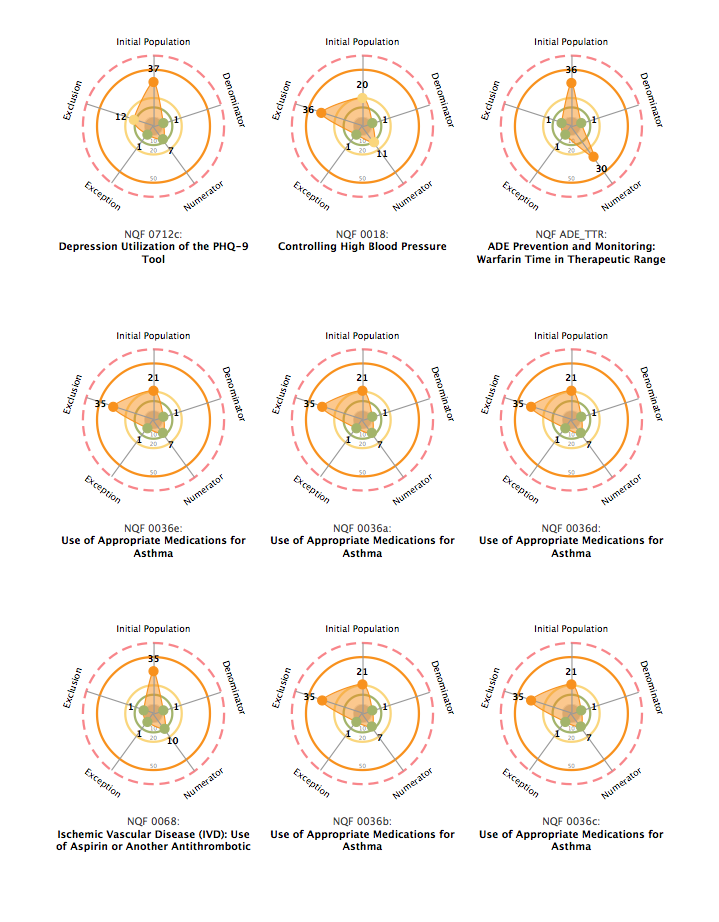
Based on our findings, MITRE is exploring the introduction of tools developed on the Kamira project that will simplify the collection and display of Cyclomatic Complexity metrics as part of the CQM development process. To see stronger adoption of this work, MITRE is ensuring that the Cyclomatic Complexity collection for CQMs is integrated data expressed in nationally recognized data standards for expressing CQMs, including the HQMF XML standard, as well as the Quality Data Model (QDM) which is used as the foundational infrastructure for building CQM logic. This close integration with existing standards is possible, and could have a large impact on the transformative impact that CQMs could have.

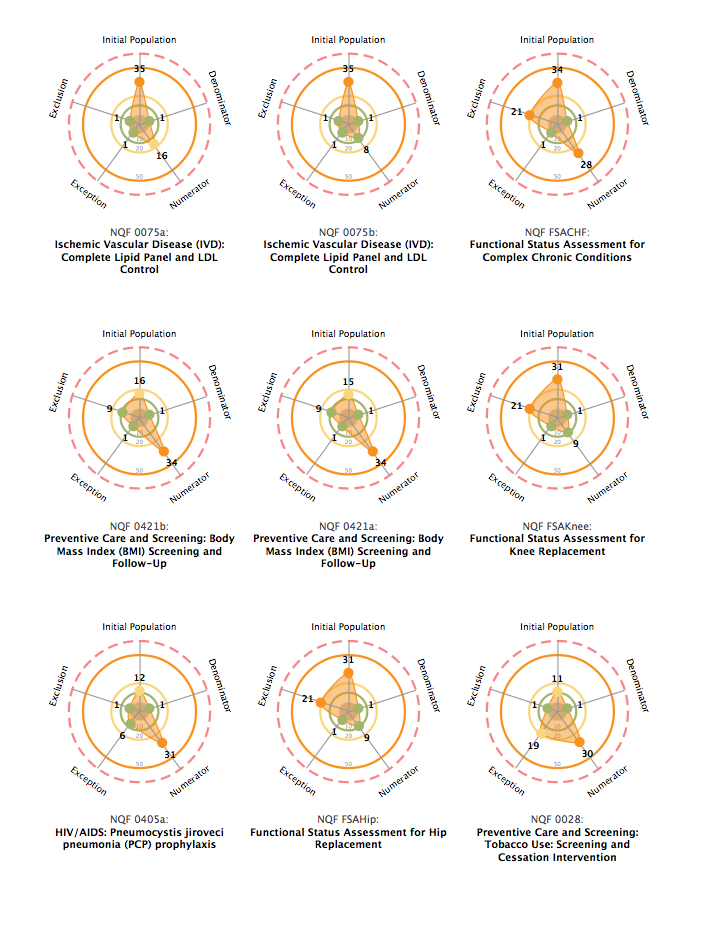
## Meaningful Use Stage 2 Ambulatory Clinical Quality Measure Complexity Evaluation

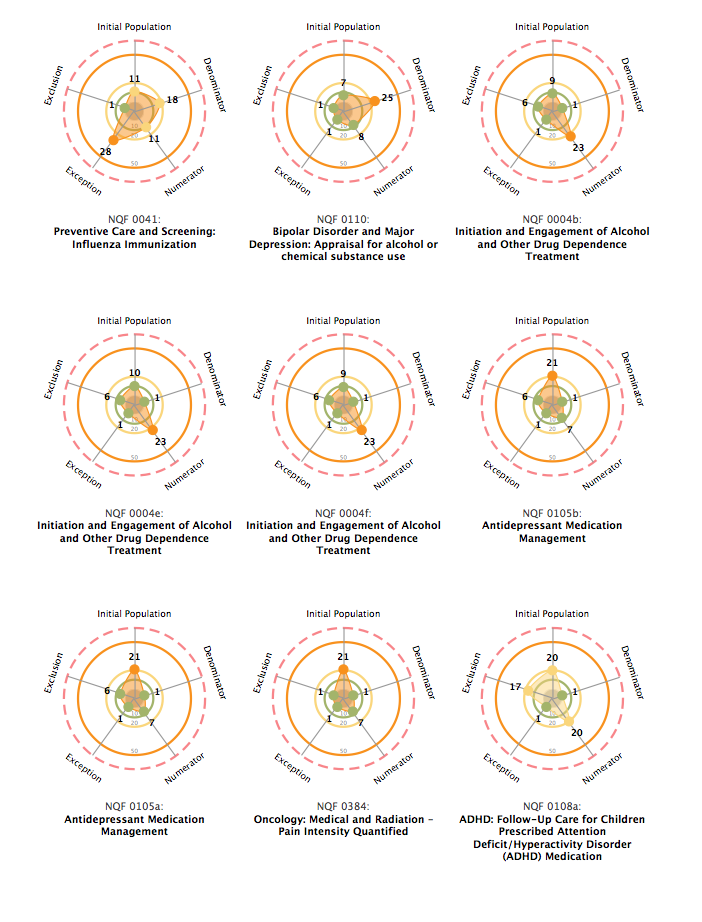


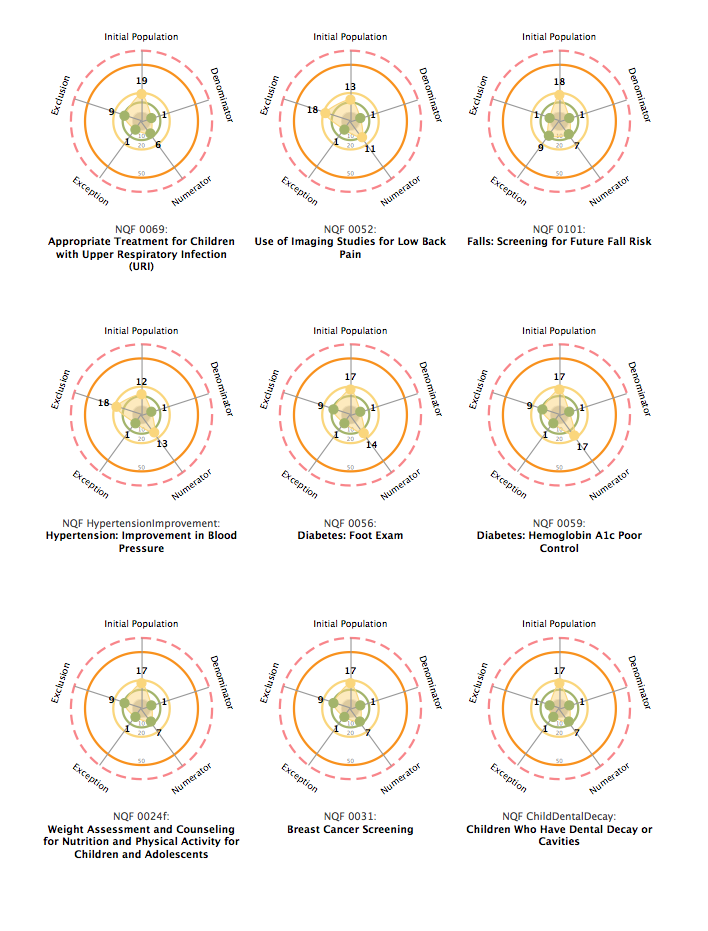


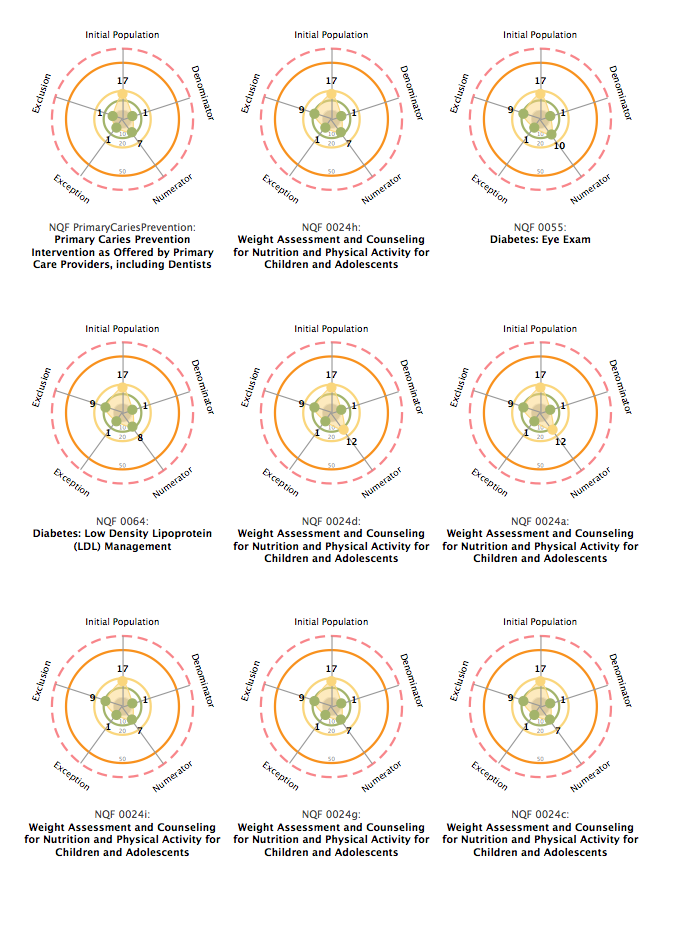




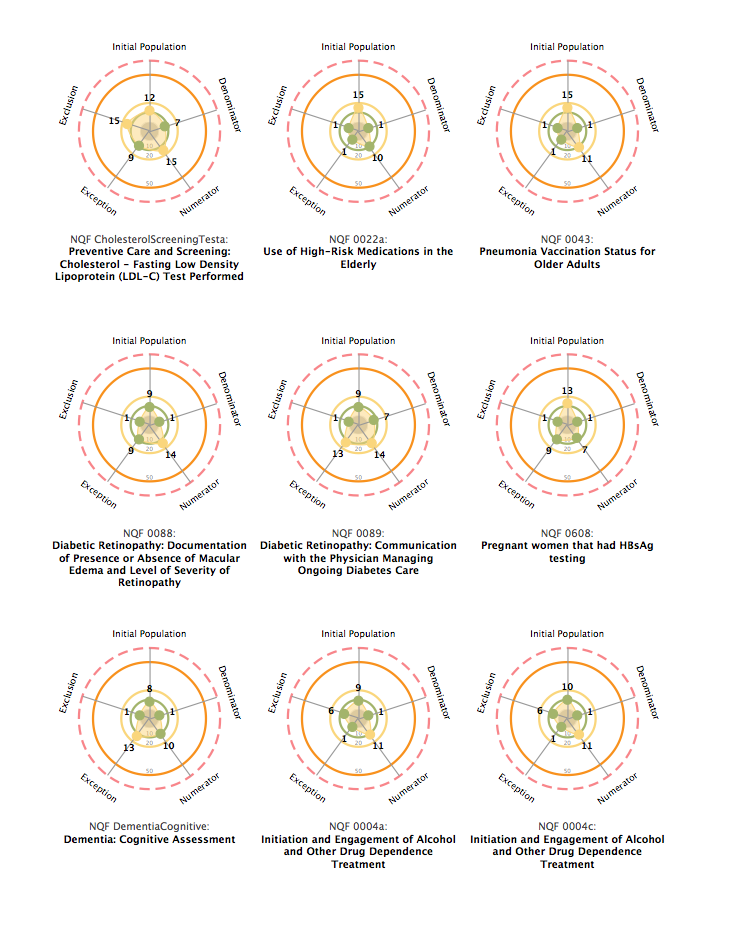


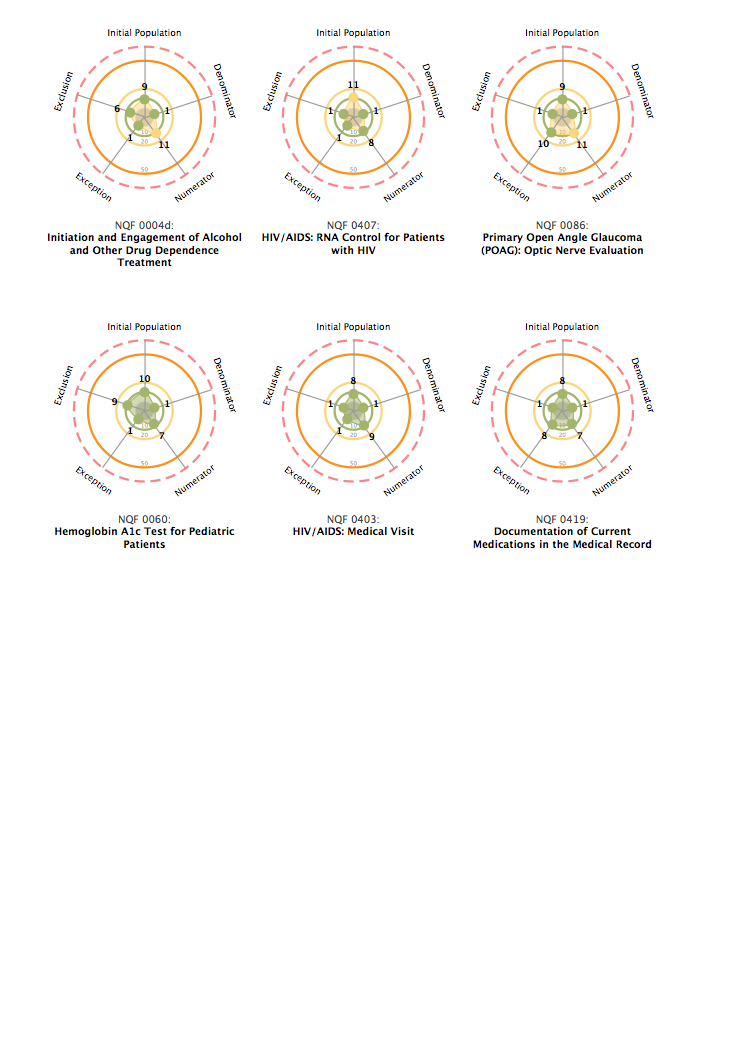












1. A Complexity Measure, Thomas J. McCabe, http://www.literateprogramming.com/mccabe.pdf [↑](#footnote-ref-1)
2. http://www.sei.cmu.edu/reports/97hb001.pdf [↑](#footnote-ref-2)
3. http://hissa.ncsl.nist.gov/sw\_assurance/strtest.html [↑](#footnote-ref-3)
4. http://projectpophealth.org/ [↑](#footnote-ref-4)
5. http://code.google.com/p/hqmf/ [↑](#footnote-ref-5)
6. http://code.google.com/p/jsmeter/ [↑](#footnote-ref-6)