**Technical Evaluation for EHR tools**

Framework for evaluation of Electronic Health Record open source tools. A reference implementation will be created so create a standard specification for other organizations. A scorecard will be developed to analyze the following tools:

* ClearHealth/MirrorMed
* Open MRS
* Tolven EHR
* World VistA
* Medsphere OpenVistA
* Ultimate EMR
* Torch
* Open EHR
* Indivo Health
* Free Med
* GNU Med (Germany)
* Open EMR (2 versions - community and managed)
* OSCAR (Canada)
* PrimaCare(Malaysia)
* Caisis

| **Evaluate** | **Description** | **Priority** |
| --- | --- | --- |
| **Open Source?** | Is it a requirement for the tool to be open source? | ?? |
| **Version** |  | N/A |
| **Source Last Updated** | *Longevity* | Medium |
| **Number of active committers** | *Longevity*: If possible, determine the approximate size of the user community based on number of downloads | Medium |
| **Downloads** | If possible, determine the approximate size of the user community based on number of downloads | Medium |
| **Year started** | *Longevity*: Is it a "fly by night" tool or has it been around for years? | Medium |
| **Architecture Type** | Maintainability: Client/Server, Rich Client, Web, Web services | Medium |
| **Programming Languages** | Maintainability: Is Java preferred? | High |
| **Database** | *Maintainability*: Database Type and Version. MySQL, PostGreSQL, other open databases? | Medium |
| **Operating System** | *Maintainability*: Does it support \*nix (Mac, Linux, Unix - which flavors), Windows (which versions?). Windows 32-bit is most often used | Medium |
| **Client** | *Maintainability*: Which client tools does it support? | Medium |
| **Coding standards** | *Maintainability*: Run against Sun standards or other (naming, etc.) | Medium |
| **Code coverage (of tests)** | *Maintainability*: Line and Path (static) | Medium |
| **Cyclomatic Complexity** | *Maintainability*: Pockets of high cyclomatic complexity. More complex code correlates to more errors. | High |
| **Abstraction** | *Maintainability* | Medium |
| **Instability** | *Maintainability* | Medium |
| **Code smells** | *Maintainability*: Large class, long method, Nested Depth, etc. | High |
| **Security Analysis** | *Reliability/Privacy*: SQL injection, Cross-site scripting | High |
| **Does the software build?** | *Ease of Use*: Is it possible to create the software? Are they using Repository pattern? | Medium |
| **Usage Documentation** | *Ease of Use*: Is the documentation to use the software easy to use? | Medium |
| **Adherence to Standards** | *Standardization*: HL7, OpenEHR , etc. | Medium |
| **Number of concurrent users** | Scalability | Medium |
| **API** | Ability to expose or accept messages/method in a standard manner | Medium |

**Potential Evaluation Tools**

* CheckStyle
* PMD
* SourceMonitor
* FindBugs
* FxCop
* Laika (EHR Testing Framework)
* Simian
* Stelligent custom static analysis tools (static test coverage and other)