Note: consider some metrics for normal software, and some which analyze open source software

Things to consider:

* Well-designed
* Well coded
* Well tested
* Error free

Open Source Factors

* Different levels of programmers (levels of involvement)
* Frequent beta releases
* Platform-independence
* Many compile-time and run-time configurations

Models: aggregate the attributes of quality in order to give an overall view of the quality

* ISO 9126 Model (not necessarily direct measurements)
  + Functionality
  + Reliability
  + Usability
  + Maintainability
  + Portability
  + Efficiency

Things to Look Over

* <http://dl.acm.org.libproxy.auc.ca/citation.cfm?id=2641525&CFID=931840923&CFTOKEN=96999064>
* <http://www.sciencedirect.com/science/article/pii/S1571066109000632>
* <http://www.sciencedirect.com/science/article/pii/S2212667813000178>
* <http://dl.acm.org.libproxy.auc.ca/citation.cfm?id=2791466&CFID=931840923&CFTOKEN=96999064>
* <http://dl.acm.org.libproxy.auc.ca/citation.cfm?id=1370131&CFID=931840923&CFTOKEN=96999064>
* <http://dl.acm.org.libproxy.auc.ca/citation.cfm?id=1083268&CFID=931840923&CFTOKEN=96999064>
* <http://dl.acm.org.libproxy.auc.ca/citation.cfm?id=1852864&CFID=931840923&CFTOKEN=96999064>
* <http://dl.acm.org.libproxy.auc.ca/citation.cfm?id=2833013&CFID=931840923&CFTOKEN=96999064>
* <http://dl.acm.org.libproxy.auc.ca/citation.cfm?id=2501539&CFID=931840923&CFTOKEN=96999064>
* <http://dl.acm.org.libproxy.auc.ca/citation.cfm?id=3011786&CFID=931840923&CFTOKEN=96999064>
* <http://flosshub.org/sites/flosshub.org/files/HalloranScherlis.pdf>

Static Errors for code quality

Coupling for code quality

Runtime Length for common tasks comparison

Documentation per code comparison

Bug resolving rate

Get a couple open source programs in same language and compare based on review