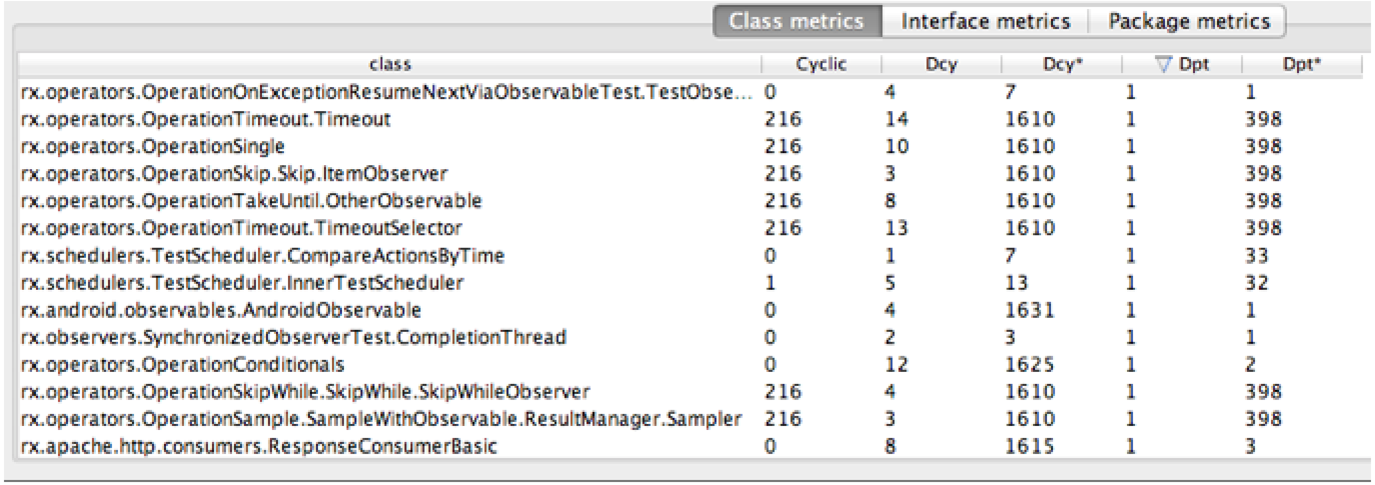
Code Metrics



class	Cyclic	Dcy	Dcy*	▽ Dpt	Dpt*
rx.operators.OperationOnExceptionResumeNextViaObservableTest.TestObse...	0	4	7	1	1
rx.operators.OperationTimeout.Timeout	216	14	1610	1	398
rx.operators.OperationSingle	216	10	1610	1	398
rx.operators.OperationSkip.Skip.ItemObserver	216	3	1610	1	398
rx.operators.OperationTakeUntil.OtherObservable	216	8	1610	1	398
rx.operators.OperationTimeout.TimeoutSelector	216	13	1610	1	398
rx.schedulers.TestScheduler.CompareActionsByTime	0	1	7	1	33
rx.schedulers.TestScheduler.InnerTestScheduler	1	5	13	1	32
rx.android.observables.AndroidObservable	0	4	1631	1	1
rx.observers.SynchronizedObserverTest.CompletionThread	0	2	3	1	1
rx.operators.OperationConditionals	0	12	1625	1	2
rx.operators.OperationSkipWhile.SkipWhile.SkipWhileObserver	216	4	1610	1	398
rx.operators.OperationSample.SampleWithObservable.ResultManager.Sampler	216	3	1610	1	398
rx.apache.http.consumers.ResponseConsumerBasic	0	8	1615	1	3

SWEN-261
Introduction to Software Engineering

Department of Software Engineering
Rochester Institute of Technology

A metric is not just a number.

- A metric is a quantitative function that calculates some characteristic and produces a numeric measurement which will be used to make a decision.
- For software product development, metrics fall into three broad categories
  - ***Process – measurements of the software process that apply across projects***
  - ***Project – measurements of one project team's activities***
  - ***Product – measurements of the resulting software product***

Software product metrics fall into multiple categories that look at different characteristics.

- Complexity
  - ***Lines of Code is the most familiar***
  - ***Cyclomatic Complexity***
- Coupling and Dependency
  - ***Robert Martin Package Metrics***
- Counting/averaging lots of things that can be counted/measured
  - ***Average lines per method***
  - ***Average parameters per method***
  - ***Average number of methods per class***
- Some metrics will apply at multiple levels, such as project, package, class, or method

Even though you can count something, it does not necessarily count for anything.

- A metric is only as good as the decisions that it will be used to make.
- Measuring something without it having a solid connection to possibly improving what you are doing is a waste of time and resources.
- Target values for measurements should be set based on a record of past measurements and resulting performance.
  - ***Measurement not in some range → some project quality was poorer***
- Initially, measurements need to be made to find the correlations.

A metric target is not absolute.

- A measurement falling outside of a target range is not an absolute indictment.
- Measurements that do not fall in the target range indicate a place for additional scrutiny.
  - *For product metrics, they indicate possible "code smells".*
  - *Places to consider for refactoring, redesign, or reimplementation*

These are some of the more popular metrics for object-oriented software systems.

- Cyclomatic complexity
  - ***Count of execution paths through a method***
- Chidamber and Kemerer
  - ***Coupling between object classes***
  - ***Lack of cohesion in methods***
- Martin Package Metrics
  - ***Fan-out coupling – classes need something outside package***
  - ***Fan-in coupling – classes outside package use something inside package***
  - ***Instability – ratio of fan-out to fan-out + fan-in***
  - ***Abstractness – ratio of abstract classes and interfaces to total number in package***

Software Engineering
Rochester Institute of Technology