

When evaluating a design, how do you evaluate complexity? [closed]

Asked 16 years, 1 month ago Modified 16 years, 1 month ago

Viewed 628 times



6



Closed. This question is [opinion-based](#). It is not currently accepting answers.



Want to improve this question? Update the question so it can be answered with facts and citations by [editing this post](#).

Closed 6 years ago.

[Improve this question](#)

We all know to keep it simple, right?

I've seen complexity being measured as the number of interactions between systems, and I guess that's a very good place to start. Aside from gut feel though, what other (preferably more objective) methods can be used to determine the level of complexity of a particular design or piece of software?

What are YOUR favorite rules or heuristics?

language-agnostic

architecture

complexity-theory

Share

Improve this question

Follow

asked Oct 31, 2008 at 14:55



Maxam

4,053 ● 2 ● 26 ● 25

7 Answers

Sorted by:

Highest score (default)



3



Here are mine:

1) How hard is it to explain to someone who understands the problem but hasn't thought about the solution? If I explain the problem to someone in the hall (who probably already understands the problem if they're in the hall) and can explain the solution, then it's not too complicated. If it takes over an hour, chances are good the solution's overengineered.

2) How deep in the nested functions do you have to go? If I have an object which requires a property held by an object held by another object, then chances are good that what I'm trying to do is too far removed from the object itself. Those situations become problematic when trying to make objects thread-safe, because there'd be many objects of varying depths from your current position to lock.

3) Are you trying to solve problems that have already been solved before? Not every problem is new (and

some would argue that none really are). Is there an existing pattern or group of patterns you can use? If you can't, why not? It's all good to make your own new solutions, and I'm all for it, but sometimes people have already answered the problem. I'm not going to rewrite STL (though I tried, at one point), because the solution already exists and it's a good one.

Share Improve this answer
Follow

answered Oct 31, 2008 at 15:03



mmr

14.9k ● 29 ● 97 ● 148



3

Complexity can be estimated with the **coupling** and how **cohesive** are all your objects. If something is have too much coupling or is not enough cohesive, than the design will start to be more complex.



Share Improve this answer
Follow

answered Oct 31, 2008 at 15:07



Patrick Desjardins

141k ● 89 ● 294 ● 346



3

When I attended the Complex Systems Modeling workshop at the New England Complex Systems Institute (<http://necsi.org/>), one of the measures that they used was the number of system states.



For example if you have two nodes, which interact, A and B, and each of these can be 0 or 1, your possible states are:



A	B
0	0
1	0
0	1
1	1

Thus a system of only 1 interaction between binary components can actually result in 4 different states. The point being that the complexity of the system does not necessarily increase linearly as the number of interactions increases.

Share Improve this answer

edited Oct 31, 2008 at 15:15

Follow

answered Oct 31, 2008 at 15:04



Ifalin

4,294 ● 5 ● 35 ● 59



1

good measures can be also number of files, number of places where configuration is stored, order of compilation on some languages.



Examples:



.- properties files, database configuration, xml files to hold related information.



.- tens of thousands of classes with interfaces, and database mappings

.- a extremely long and complicated build file (build.xml, Makefile, others..

Share Improve this answer

answered Oct 31, 2008 at 15:11

Follow



[webclimber](#)

2,640 ● 4 ● 26 ● 36

Not the complexity of the code but the design, hard to know how many file when designing ;) – [Patrick Desjardins](#) Oct 31, 2008 at 15:20



0



If your app is built, you can measure it in terms of time (how long a particular task would take to execute) or computations (how much code is executed each time the task is run).

If you just have designs, then you can look at how many components of your design are needed to run a given task, or to run an average task. For example, if you use MVC as your design pattern, then you have at least 3 components touched for the majority of tasks, but depending on your implementation of the design, you may end up with dozens of components (a cache in addition to the 3 layers, for example).

Share Improve this answer

answered Oct 31, 2008 at 15:00

Follow



[Elie](#)

13.8k ● 25 ● 78 ● 128

Finally something LOC can actually help measure? :)



i think complexity is best seen as the number of things that need to interact.

0



A complex design would have n tiers whereas a simple design would have only two.



Complexity *is* needed to work around issues that simplicity cannot overcome, so it is not always going to be a problem.

There is a problem in defining complexity in general as complexity usually has a task associated with it.

Something may be complex to understand, but simple to look at (very terse code for example) The number of interactions getting this web page from the server to your computer is very complex, but the abstraction of the http protocol is very simple.

So having a task in mind (e.g. maintenance) before selecting a measure may make it more useful. (i.e. adding a config file and logging to an app increases its objective complexity [yeah, only a little bit sure], but simplifies maintenance).

[Share](#) [Improve this answer](#)

[edited Oct 31, 2008 at 19:27](#)

[Follow](#)

answered Oct 31, 2008 at 19:08



[Nat](#)

14.3k ● 5 ● 43 ● 64



There are formal metrics. Read up on [Cyclomatic Complexity](#), for example.

-1



Edit.



Also, look at [Function Points](#). They give you a non-gut-feel quantitative measurement of system complexity.

Share Improve this answer

edited Oct 31, 2008 at 19:03

Follow

answered Oct 31, 2008 at 15:29



[S.Lott](#)

391k ● 82 ● 517 ● 788

cyclomatic complexity is a mteric computed from the source code - you don't have the source code during design!

– [Steven A. Lowe](#) Oct 31, 2008 at 15:36

@Steven A. Lowe: You do, however, have the same concepts of if-statements. You can easily score parts of your design with complexity based on choices and decisions described in the specifications or requirements. – [S.Lott](#) Oct 31, 2008 at 15:38

Cyclomatic is better when you have the source code. In design it's simply not productive to try to estimate the number of loop and comparaison. – [Patrick Desjardins](#) Oct 31, 2008 at 15:51