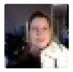
How do you explain Separation of Concerns to others?

Ask Question

If you had a colleague who didn't understand the benefits of Separation of Concerns, or didn't understand it quite enough to code-quality object-oriented

asked Dec 30 '10 at 12:18



Marcie
2,399 2 13 21

Home

Questions

Tags

Users

Unanswered

5 Answers

Imagine you have a program which has been released. A customer comes along and offers to pay you for a enhancement to one of its features. In order to get the money, you will need to change your program to add the new feature. Some of the things that will influence what your profit margin is are:

1. how much code you have to change
2. how easy it is to make the changes
3. how likely you are to break existing features that are being used by other customers
4. how much you can reuse you existing model/architecture

Separation of concerns helps you to get more positive answers to these questions.

1. if all of the code for a particular behaviour of the application is separated out, then you will only have to change code directly associated with your new feature. Which should be less code to change.
2. if the behaviours you are interested in are neatly separated from the rest of the application it is more likely you will be able to swap in a new implementation without having to fully understand or manipulate the rest of the program. It should also be easier to find out which code you need to

3. Code that you do not have to change is less likely to break than code that you do change. So splitting up the concerns helps you to avoid breakage in unrelated features by preventing you from having to change code that they could call. If your features are mixed up together you might change the behavior of one by accident while trying to change another one.
4. If your architecture is agnostic to technical or business logic detail then changes to implementation are less likely to require new architectural features. For example, if your main domain logic is database agnostic then supporting a new database should be as easy as swapping in a new implementation of the persistence layer.

edited Dec 30 '10 at 18:07

answered Dec 30 '10 at 14:38

flamingpenguin
2,416 10 12

I love that you made the answer firmly anchored in the financial reality. Managers have no excuse to be sloppy and ignore this fundamental concept. – moodboom Dec 10 '16 at 16:21

Look at a hospital, and think about all of the different roles that are involved in providing care to a patient: triage nurses, Is there any one person that knows how *all* of those people get their jobs done? No, because it would be overwhelming. They have to separate out the different responsibilities into distinct roles and the touchpoints between those roles are very specific.

edited Sep 2 '12 at 20:35
user34530

answered Dec 30 '10 at 13:26
RationalGeek
7,709 6 33 53

If he/she works in an office, take it as an example, explain the role of each staffs in that office, and ask him, what would happen, if those staffs aren't divided according to their jobs?

answered Dec 30 '10 at 12:48
Abimaran Kugathasan
953 1 12 23

I would look at how he failed to apply SoC in his code/design and turn that into a real-world example that he can relate with and that is obviously undesired.

For example, if he has a class where the client needs to supply several pieces of information that are not relevant for those clients, then I would use the analogy of a bakery where you have to bring your own grains and yeast if you want to buy a bread.

answered Dec 30 '10 at 14:22
Bart van Ingen Schenau

One example might be a html developer might want to separate out html, css and javascript into separate files. This way you can change the look and feel of something say by simply modifying the css or the behaviour of something by changing the javascript file that is loaded separately. If you have a responsive or adaptive site this paradigm works well as you can load different css or javascript depending on a users viewport or user agent. However if you modify the html or template, chances are that either the css or javascript could break. These separate concerns can also be dependent.

Another approach is to bundle all your css javascript and html into a group of components or modules. This means that you can make changes to one module and it should not affect other components or modules on the page that it runs along side of that are not related. Here the css, js and html files are merged into a single component that can be unit tested. So the separation of concerns comes in the form of individual atomic components that can be unit tested rather than separation of markup, styling and behavioural elements. This second approach is more suited to creating more complex web applications.

edit. Since I have received negative response to this comment I thought I would revisit it and try to qualify some of my pov. Unfortunately any feedback here is not particularly constructive but I did see an interesting discussion elsewhere that looks at React, the current hot technology in web development, a real world example, and asks if it breaks separation of concerns or in particular if it breaks one of the principle's of Feather's SOLID object orient design methodology.

The technical JavaScript Developer Perspective

```
NO, because JSX is a view language. That's one responsibility.
BUT, this implies that the JS developer is self-enforcing SoC/SR
architecture by not mixing ViewModel concerns in his JSX. This t
```

The UX/UI Designer Perspective

```
YES, because JSX mixes Semantic Content (Model) with Behavior (C
YES, because the intrusion, specifically of JavaScript, into the
difficult or impossible for me to play my role and leverage my e
```

The Team Perspective

```
NO, if both...
Separate files are used for the View (JSX) and ViewModel (JS).
Either there aren't UI/UX/Designers involved, or they are produc
with JSX (not very common).
YES, if either...
Everything is in the same file, causing problems for version con
modern editors.
Members of the team who are comfortable with HTML/CSS but less c
are excluded because of mixture or roles.
```

https://hashnode.com/post/does-react-really-violate-separation-of-concern-by-putting-html-and-js-in-a-single-file-cil3bn5hj0011a65347rsdut0

Also on the page is a link to an interesting presentation from Pete Hunt, of Facebook, where he talks about components not templates, and separating out the concerns in the language application rather than separating out concerns of the framework, i.e. templates, css and javascript etc.

With regards separating out your concerns in the language of your application this could involve use various patterns to separate out or decouple your code into modular form that can be unit tested etc.

So to summarise, separating out concerns can depend on your role or point of view, as mentioned else where.

edited Jan 24 '17 at 19:36

109 3

1 this doesn't seem to offer anything substantial over points made and explained in prior 7 answers – gnat Jan 20 '17 at 21:43

I am just pointing out that separating out concerns can take different approaches depending on the context. This is closer to a real world situation in terns of software engineering and I am highlighting that there are different approaches you can take when working on html pages that may at first seem contradictory. – dan Jan 20 '17 at 22:12

protected by gnat Jan 20 '17 at 21:42

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 reputation on this site (the association bonus does not count).

Would you like to answer one of these unanswered questions instead?