

Design Defects and Restructuring

Engr. Abdul-Rahman Mahmood



abulrahman@nu.edu.pk



alphapeeler.sf.net/pubkeys/pkey.htm



pk.linkedin.com/in/armahmood



www.twitter.com/alphapeeler



www.facebook.com/alphapeeler



abulmahmood-sss



alphasecure



armahmood786



http://alphapeeler.sf.net/me



alphapeeler#9321



reddit.com/user/alphapeeler



www.flickr.com/alphapeeler



http://alphapeeler.tumblr.com



armahmood786@jabber.org



alphapeeler@aim.com



mahmood_cubix



48660186



alphapeeler@icloud.com



pinterest.com/alphapeeler



www.youtube.com/user/AlphaPeeler

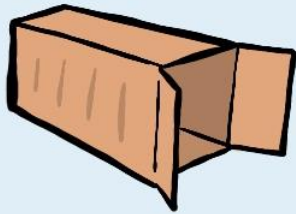
Design Principle

Encapsulate that varies

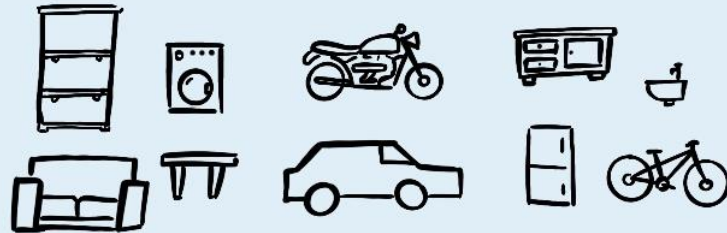
Encapsulate What Varies

- Encapsulate What Varies, or 'Encapsulate What Changes' is the technique of reducing the impact of frequently changing code by encapsulating it.

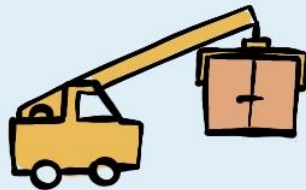
ENCAPSULATING



SOMETHING THAT CHANGES FREQUENTLY




REDUCES THE IMPACT ON WHAT INTERACTS WITH IT



- **Encapsulating what varies** is a technique that helps us handle frequently changing details. Code tends to get tangled when it is continuously modified due to new features or requirements. By isolating the parts which are prone to change we limit the surface area that will be affected by a shift in requirements.

Example 1

```
// ✗ This is hard to understand and subject to change.  
// We may need to check if a book is reserved.  
function checkoutBook(customer, book) {  
  if (  
    customer &&  
    customer.fine <= 0.0 &&  
    customer.card &&  
    customer.card.expiration === null &&  
    book &&  
    !book.isCheckedOut  
  ) {  
    customer.books.push(book)  
    book.isCheckedOut = true  
  }  
  return customer  
}
```



```
// ✓ This is easy to read and won't change even if the
checkout requirements vary.
function checkoutBook(customer, book) {
  if (customer.canCheckout(book)) {
    customer.checkout(book)
  }

  return customer
}
```

Example 2

```
if (pet.type() == dog) {  
    pet.bark();  
} else if (pet.type() == cat) {  
    pet.meow();  
} else if (pet.type() == duck) {  
    pet.quack()  
}
```

or you can write code that looks like this:

```
pet.speak();
```