

what is preventing me
to write nearly
accurate code

nhm tanveer hossain khan (hasan)
<http://hasan.we4tech.com>
somewhere in...

what are we going to talk today?

- what is preventing me (probable you too)
- how to write nearly accurate code
- question / answer session

what is preventing me?
my observation

being too honest!

- in commitment
- in writing code
- skipping known issue
- not feeling colorful moment
- not feeling togetherness
- not feeling inclusiveness
- not feeling in team way instead of my way.
- focusing on too many projects

! patience

- from company
- from me
 - => spaghetti code
 - => quick solution

unclear understanding of requirements

- from client
- from product owner
- from user story or use case
- or too detailed spec (feel like class book)

commitment overflow

- from client
- from product owner
- from peer

not enough space for
refactoring

lack of domain specific
knowledge

over complicating code/ architecture

- not mockable environment
- not testable environment
- bootstrapping takes longer than 1 minute
- \leq usually team does
 - work with workaround
 - re-code similar function

most smart code

most smart code

```
def test_limited_list_of_requests  
  get(:requests)  
  # ... rest goes here....  
end
```

most smart code

```
def self.included(base)
  %w( get post put delete head ).each do |method|
    base.class_eval <<-EOV, __FILE__, __LINE__
      def #{method}(action, parameters = nil, session = nil, flash = nil)
        process(action, parameters, session, flash)
      end
    EOV
  end
end
```

break!

take deep breath !!

welcome back!!!

how to write nearly
accurate code

patience

- being honest
- understanding ethics
- feel in team way
- document all known issues
- (example) formatted user story
 - **As a** user
I want to add comment
So that visitor can find my understanding.
- comment everything
- follow coding convention

think some one
watching your code !!

behavior driven development (BDD)

- before development
 - create spec
 - verify spec with stakeholder
- during development
 - ensure all spec passed
 - ensure all spec expectations were met

BDD example

BDD example

```
context Company, "while creating a new company" do  
  it "should not be valid without a company name" do  
    @company.should_not_be_valid  
    @company.errors.on(:name).should == "can't be blank"  
  end  
  
  it "should not have error after save" do  
    @company.name = "somewhere in..."  
    @company.save.should_be_true  
    @company.errors.should_be_empty  
  end  
end
```

BDD example

```
Run ▶ service_spec  
▶▶ while creating a new company  
- should not be valid without company name  
|| - should not have error after save
```

remember !!

*“your code is mirror of
you”*

nothing preventing me
to say you

nothing preventing me
to say you
thank you!!!

now what is preventing
you? (Q/A)