
SURVIVING
LEGACY {code}
IT'S (PROBABLY) NOT
AS BAD AS YOU THINK
SOFTWARE CRAFTSMANSHIP
Desire to be proud of what we do

BHARAT
MANE

prelude

WHAT IS LEGACY CODE?

wikipedia claims as the original definition

“CODE FOR AN OBSOLETE COMPUTER”

most current definition

“CODE INHERITED FROM SOMEONE ELSE”

reality check 1

I inherited some code that was not legacy code

reality check 2

I found myself writing exciting new legacy code that was legacy even before it left my fingers

most visceral definition

“CODE I AM AFRAID TO MODIFY”

more practical definition that points to a solution

“CODE WITHOUT TESTS”

— Michael Feathers

another definition that I liked

“CODE THAT I DON’T HAVE CONFIDENCE IN!”

— Werner Beytel

is there better way to look at legacy code?

WHEN I SAY “LEGACY CODE”

…I mean profitable code that we feel afraid to change.

— J. B. Rainsberger

is there better way to look at legacy code?

WHEN I SAY “LEGACY CODE”

…I mean **profitable** code that we **feel afraid** to **change**.

— J. B. Rainsberger

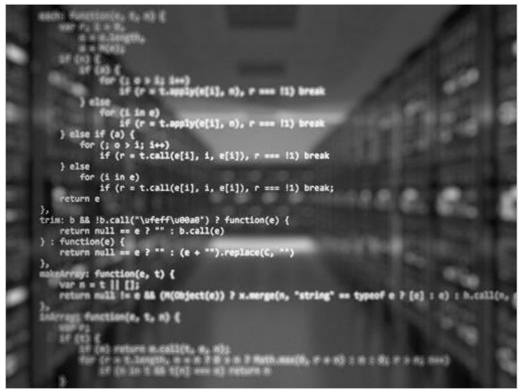
WRITE SOME TESTS

~~all done, lets go home~~

IF ONLY IT WERE THAT EASY

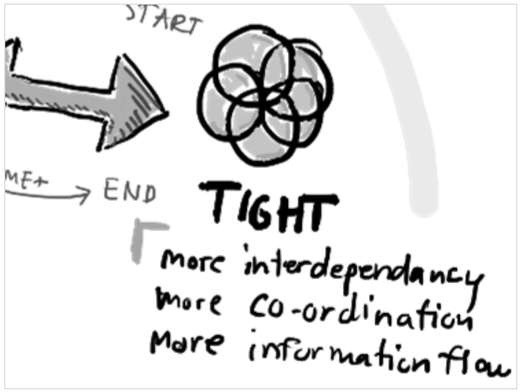
WHY IT'S NOT THAT EASY?

SPAGHETTI CODE



GIANT FUNCTIONS

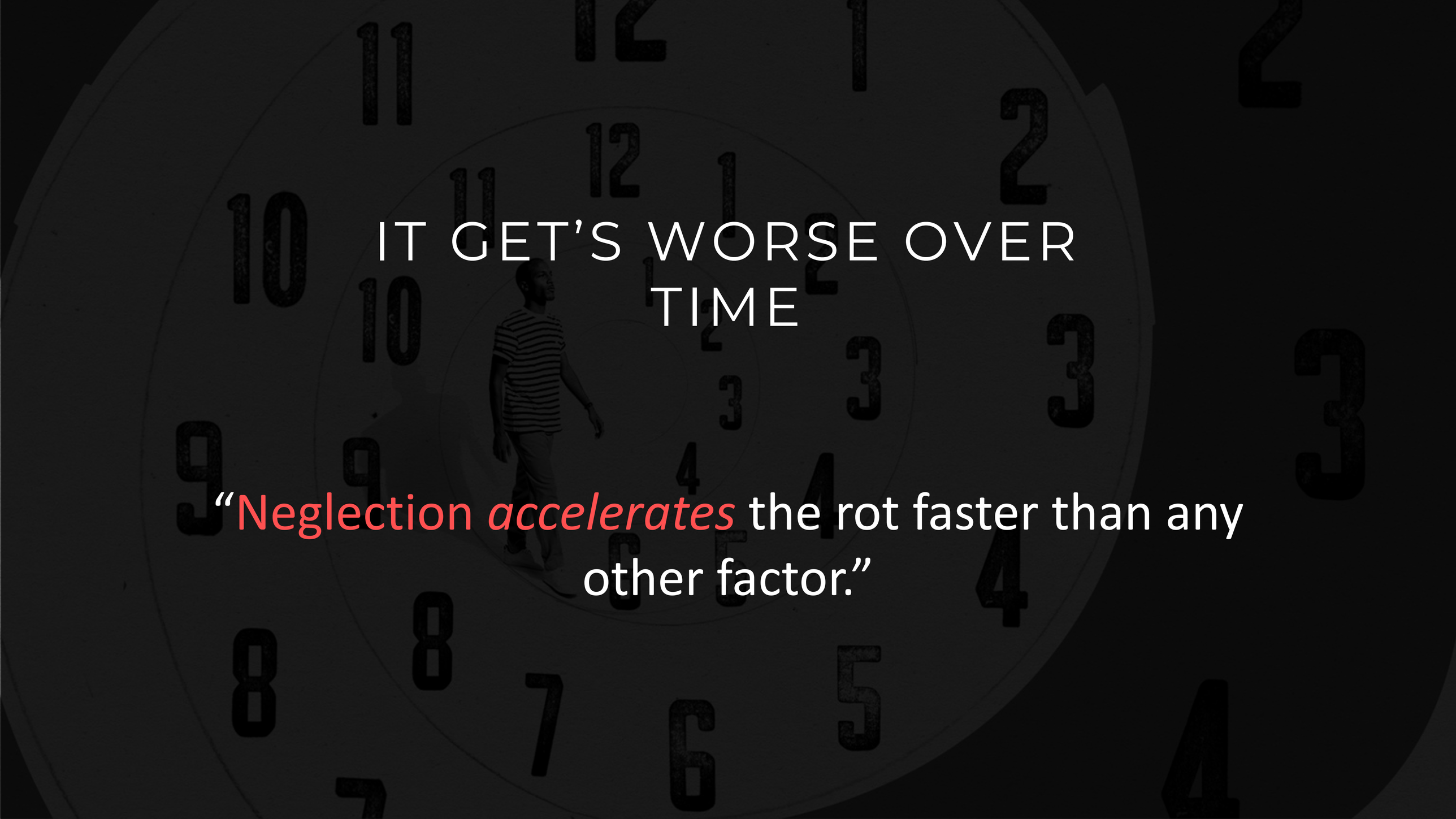
GLOBAL MUTABLE STATE



TIGHTLY COUPLED DEPENDENCIES

the central conflict of legacy code

Reference: **J. B. Rainsberger**



IT GET'S WORSE OVER TIME

"Neglection *accelerates* the rot faster than any other factor."

3 KEYS AREAS IN SURVIVING LEGACY CODE

```
in people.data.users:
esponse = client.api.statuses.user_timeline.get(screen_name=
int 'Got', len(response.data), 'tweets from', i.screen_name
 len(response.data) != 0:
  ltdate = response.data[0]['created_at']
  ltdate2 = datetime.strptime(ltdate,'%a %b %d %H:%M:%S +00
  today = datetime.now()
  howlong = (today-ltdate2).days
  if howlong < daywindow:
      print i.screen_name, 'has tweeted in the past' , dayw
      totaltweets += len(response.data)
      for j in response.data:
          if j.entities.urls:
              for k in j.entities.urls:
                  newurl = k['expanded_url']
                  urlset.add((newurl, j.user.screen_name))
  else:
      print i.screen_name, 'has not tweeted in the past', d
```

RESCUE THE CODE

ORGANIZE THE WORK

NAVIGATE THE PEOPLE

Reference: **J. B. Rainsberger**

SURVIVING LEGACY CODE IN A NUTSHELL

```
in people.data.users:
esponse = client.api.statuses.user_timeline.get(screen_name=
int 'Got', len(response.data), 'tweets from', i.screen_name
 len(response.data) != 0:
  ltdate = response.data[0]['created_at']
  ltdate2 = datetime.strptime(ltdate,'%a %b %d %H:%M:%S +00
  today = datetime.now()
  howlong = (today-ltdate2).days
  if howlong < daywindow:
      print i.screen_name, 'has tweeted in the past' , dayw
      totaltweets += len(response.data)
      for j in response.data:
          if j.entities.urls:
              for k in j.entities.urls:
                  newurl = k['expanded_url']
                  urlset.add((newurl, j.user.screen_name))
  else:
      print i.screen_name, 'has not tweeted in the past', d
```

RESCUE THE CODE

ORGANIZE THE WORK

NAVIGATE THE PEOPLE

RESCUE THE CODE

STOP MAKING IT WORSE

OPTIMIZE FOR SAFETY

RESCUE THE CODE

STOP MAKING IT WORSE

even if we don’t rescue the legacy code…

…USE THIS AS A RICH SET EXAMPLES OF WHAT WE WILL **NOT DO**

STATIC ANALYSIS TOOLS ARE YOUR FRIENDS

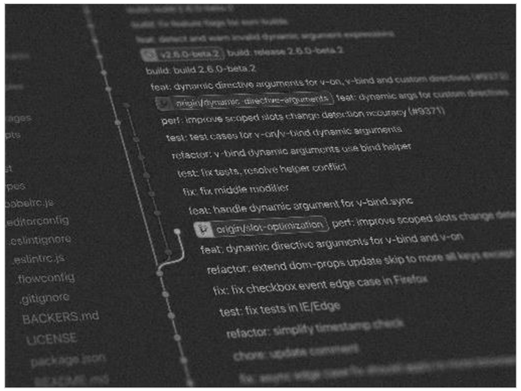
RESCUE THE CODE

OPTIMIZE FOR SAFETY

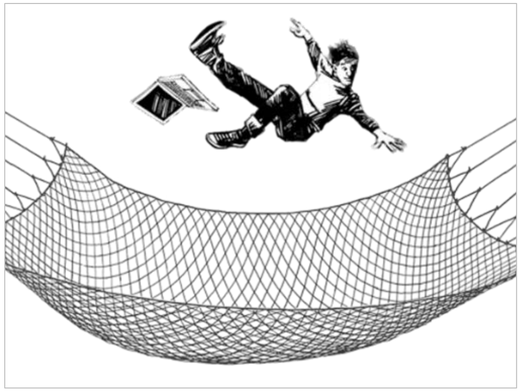
OPTIMIZE FOR SAFETY

BUILD SKILL

GET FAMILIAR



MICROCOMMITTING



SAFETY NET

gold master test

A gold master test is a regression test for complex, untested systems that asserts a consistent macro-level behavior.

let’s

{code}

https://github.com/bharatmane/trivia-refactoring/commits/master

Approval Tests

ApprovalMaintenance.VerifyNoAbandonedFiles(String[] ignore)

Approvals.Verify(FileInfo receivedFilePath)

Approvals.VerifyAll(IDictionary<K, V> dictionary)

Approvals.VerifyBinaryFile(Byte[] bytes, String fileExtensionWithDot)

Approvals.VerifyException(Exception e)

Approvals.VerifyExceptionWithStacktrace(Exception e)

Approvals.VerifyFile(String receivedFilePath)

Approvals.VerifyHtml(String html)

Approvals.VerifyJson(String json)

Approvals.VerifyPdfFile(String pdfFilePath)

Approvals.VerifyWithCallback(Object text, Action<String> callBackOnFailure)

Approvals.VerifyWithExtension(String text, String fileExtensionWithDot, Func<String, String> scrub

Approvals.VerifyXml(String xml)

Approver.Verify(IApprovalApprover approver, IApprovalFailureReporter reporter)

CombinationApprovals.VerifyAllCombinations(Func<A, B, C, D, E, F, G, H, I, Object> processCall, Func<Object, String> resultFormatter, IEnumerable<A> aList, IEnumerable<B> bList, IEnumerable<C> cList, IEnumerable<D> dList, IEnumerable<E> eList, IEnumerable<F> fList, IEnumerable<G> gList, IEnumerable<H> hList, IEnumerable<I> iList)

EmailApprovals.Verify(MailMessage email)

EmailApprovals.VerifyScrubbed(MailMessage email, Func<String, String> scrubber)

EventApprovals.VerifyEvents(Object value)

HtmlApprovals.VerifyHtml(String html, Func<String, String> scrubber)

HtmlApprovals.VerifyHtmlStrict(String html)

SerializableTheory.Verify(Object original, Action<Object, Object> assertEqual)

SetApprovals.VerifyFileAsSet(String filename)

SetApprovals.VerifySet(IEnumerable<T> enumerable, Func<T, String> formatter)

ThreadSafetyTheory.VerifyNoRaceConditions(Int32 times, Func<T> caseGenerator, Func<T, String> caseString, Func<T, Object> possibleRaceConditionFunction, Func<T, Object> knownGoodFunction)

XmlApprovals.VerifyOrderedXml(String text, Func<String, String> scrubber)

XmlApprovals.VerifyText(String text, String fileExtensionWithoutDot, Boolean safely, Func<String, String> scrubber)

XmlApprovals.VerifyXml(String xml, Func<String, String> scrubber)

tl;**dr**

more practical definition that points to a solution

“CODE WITHOUT TESTS”

— Michael Feathers

is there better way to look at legacy code?

WHEN I SAY “LEGACY CODE”

…I mean **profitable** code that we **feel afraid** to **change**.

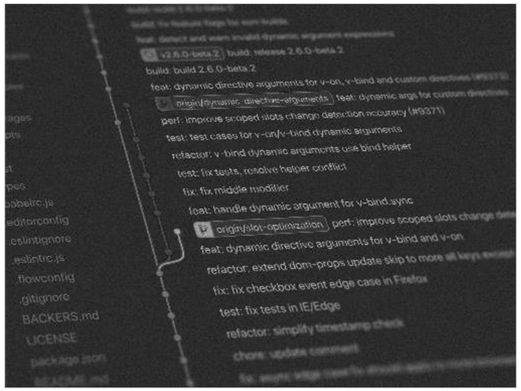
— J. B. Rainsberger

NEGLECTION **ACCELERATES** ROT FASTER THAN ANY OTHER FACTOR

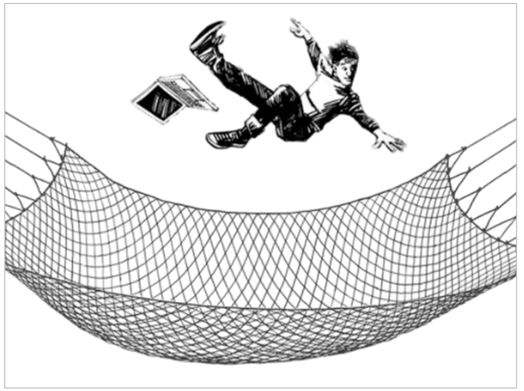
OPTIMIZE FOR SAFETY

BUILD SKILL

GET FAMILIAR



MICROCOMMITTING



SAFETY NET

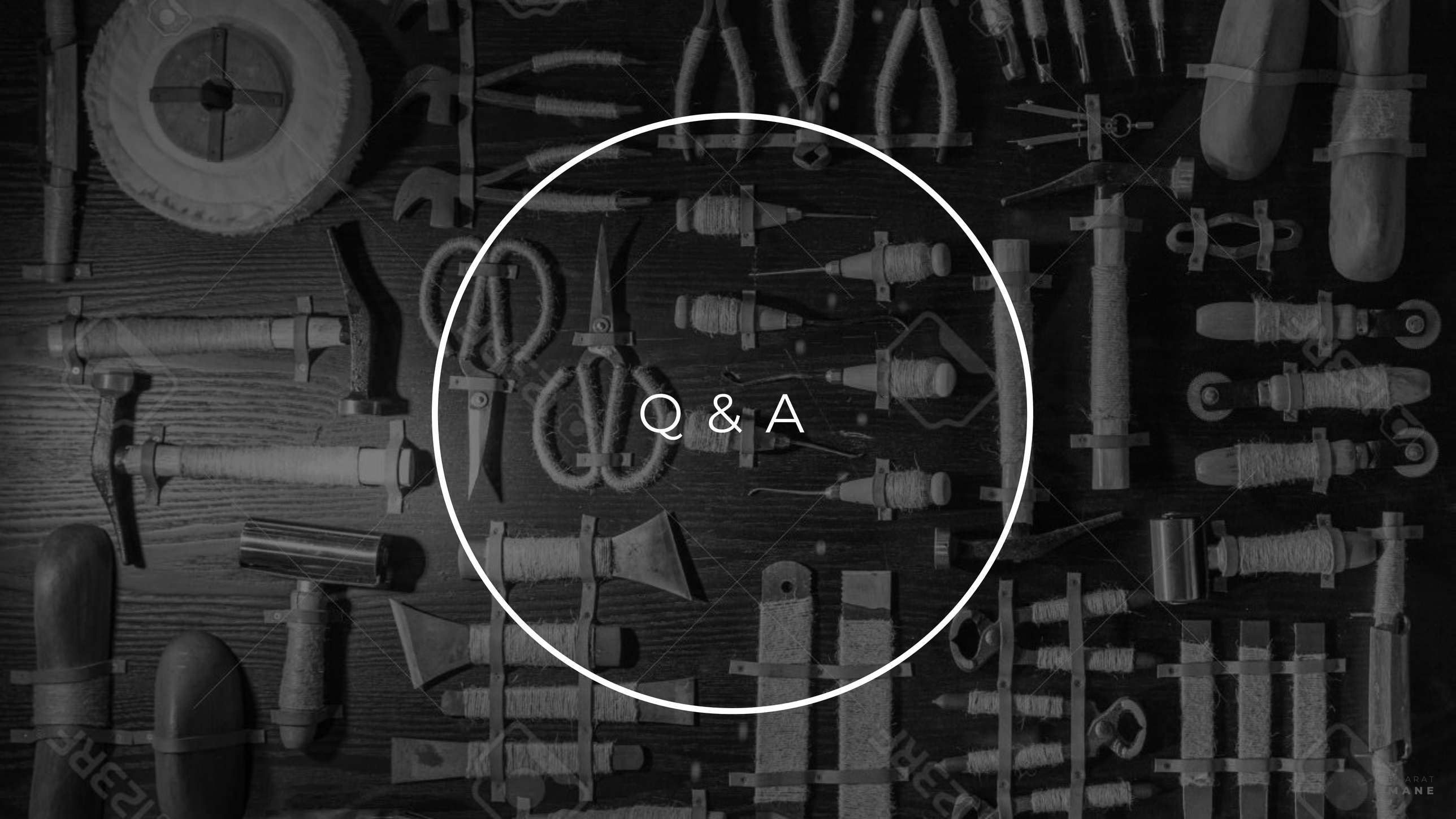
GOLDEN MASTER

Approval Tests

PRACTICE

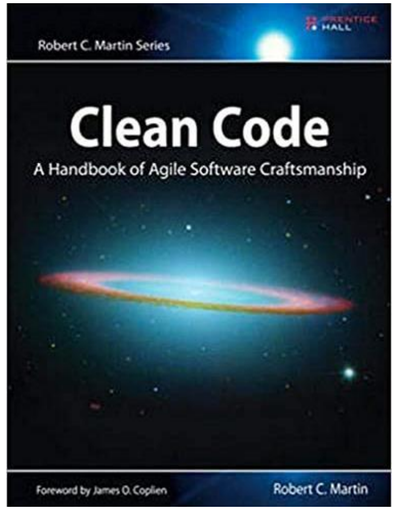
PRACTICE

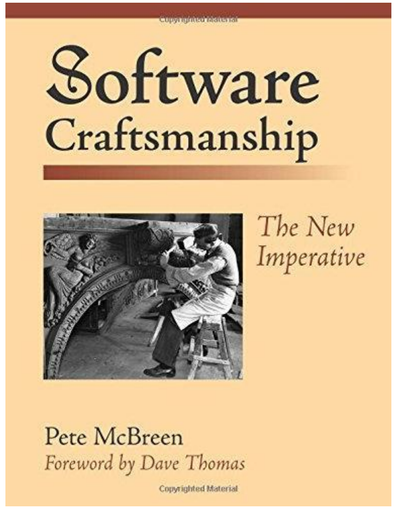
PRACTICE

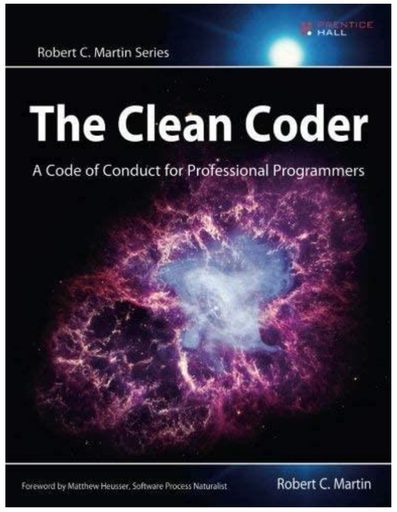

Q & A

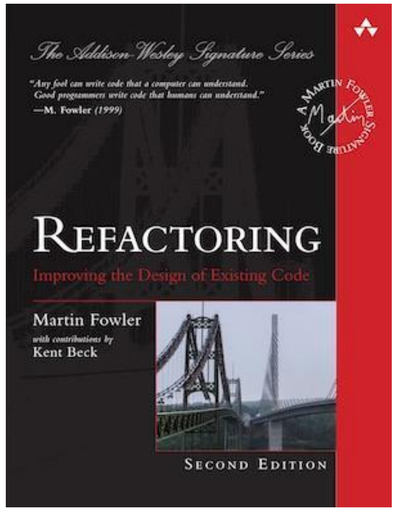
ANNEXTURE

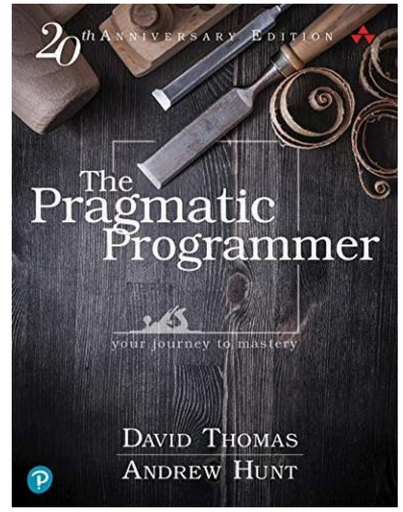
GOOD READS

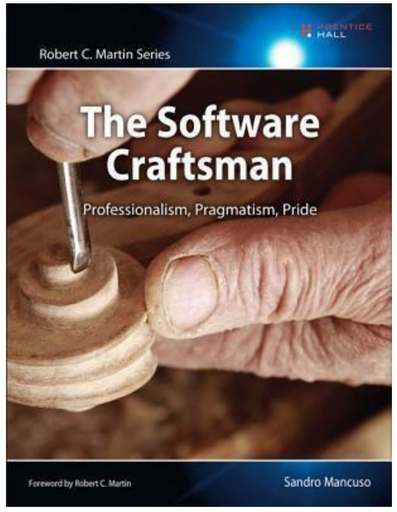


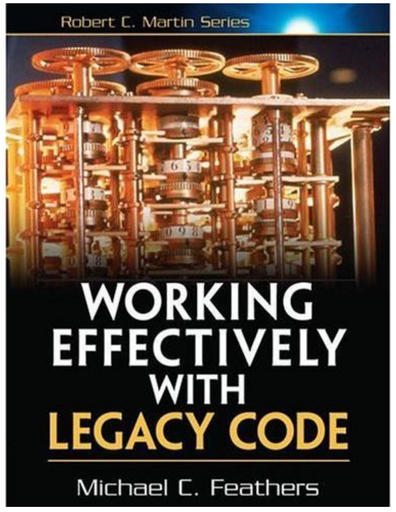


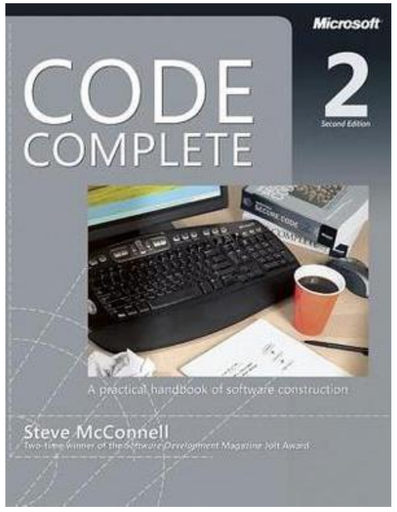


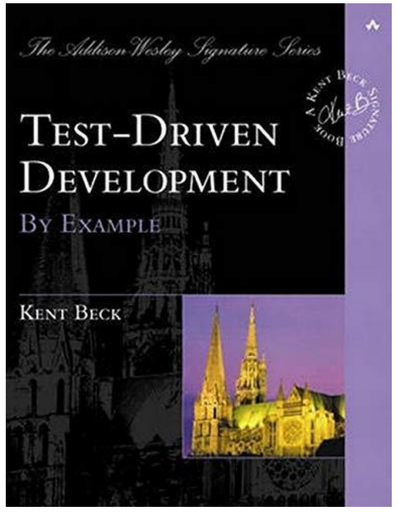


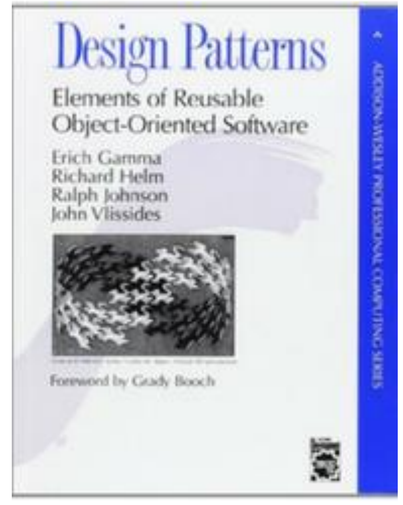






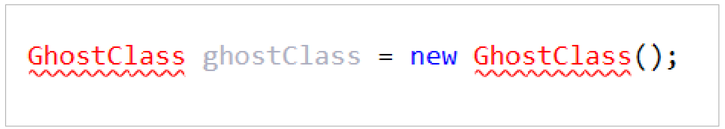


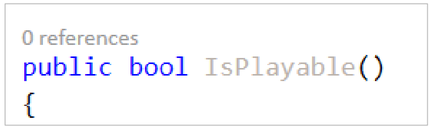




error *warn* *typo* *unused*

When it's {red}, {yellow}, {green}, {gray}





Alt + Enter will save your day