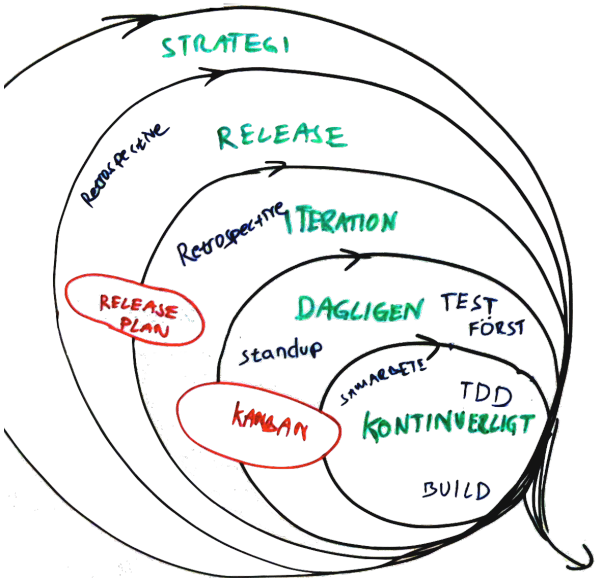
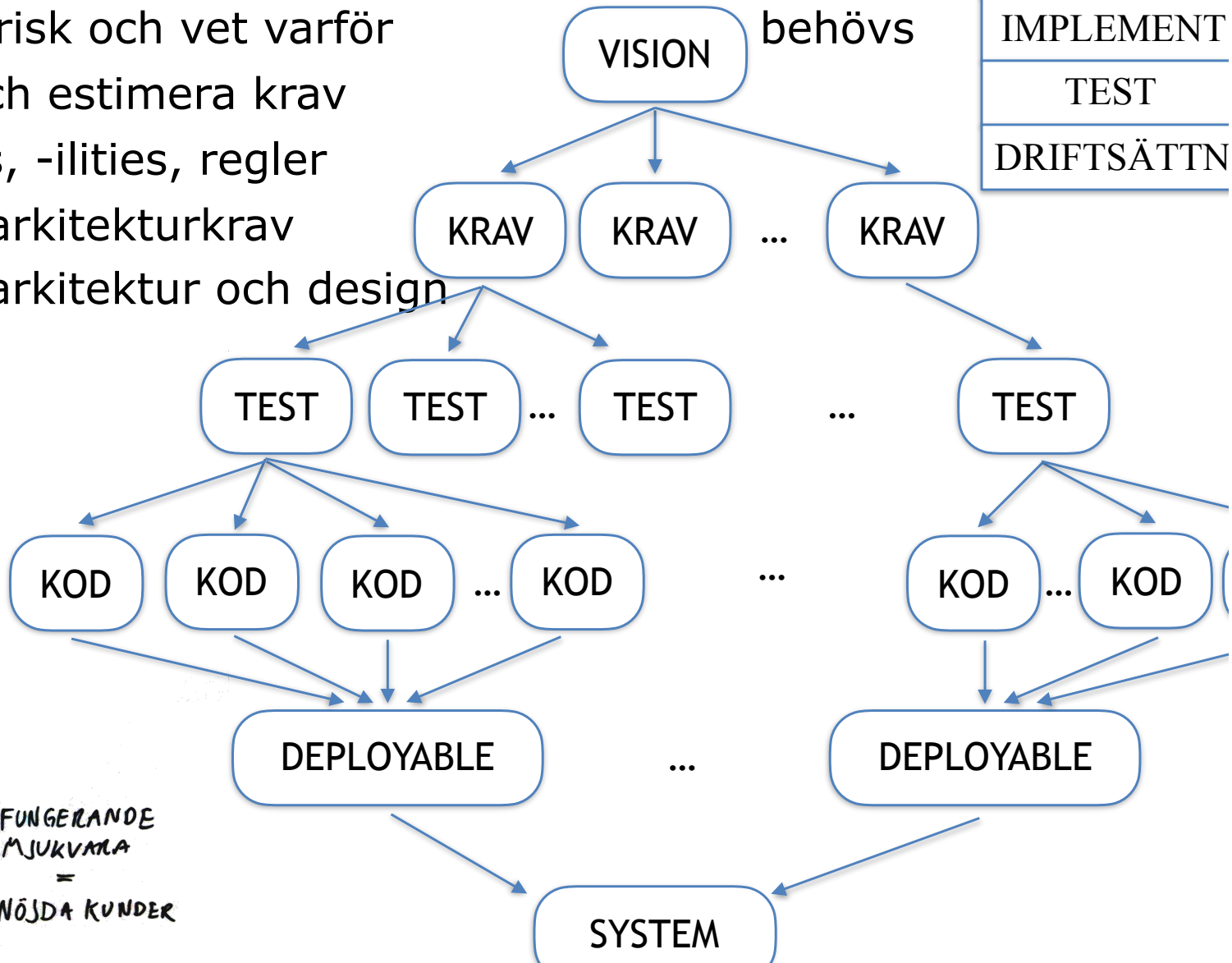
Den Röda Tråden

- Vi kan välja utvecklingsmodell
- Vi kan hantera risk och vet varför
- Vi kan skriva och estimera krav
  - User stories, -ilities, regler
- Vi kan ta fram arkitekturkrav
- Vi kan ta fram arkitektur och design



FUNGERANDE MJUKVARA = NÖJDA KUNDER



Implementation

- Röda tråden
- **Software Craftmanship**
- Clean code
- Code Quiz
- (Teststrategier)
- (Vad är en bra implementation?)

Software craftsmanship

Software craftsmanship

Sustainable pace - Cost of change - Continuous Delivery

”The crap in in your code will build up faster than you can deal with it”

Testing

”The more effort we put into caching defects early, the faster we go”

”Take less time to deliver better software”.

Automated Regression Test

Readability

KISS - Keep It Simple Stupid!

The more complicated code is the harder it is to test, the harder it is to understand, much harder to change.

Duplication multiplies the cost of change.

The ripple effect. Craftsmen manage the dependencies in their code.

”As craftsmen there are things we can do”(to cost less)”, steps we can take, skills we can apply”

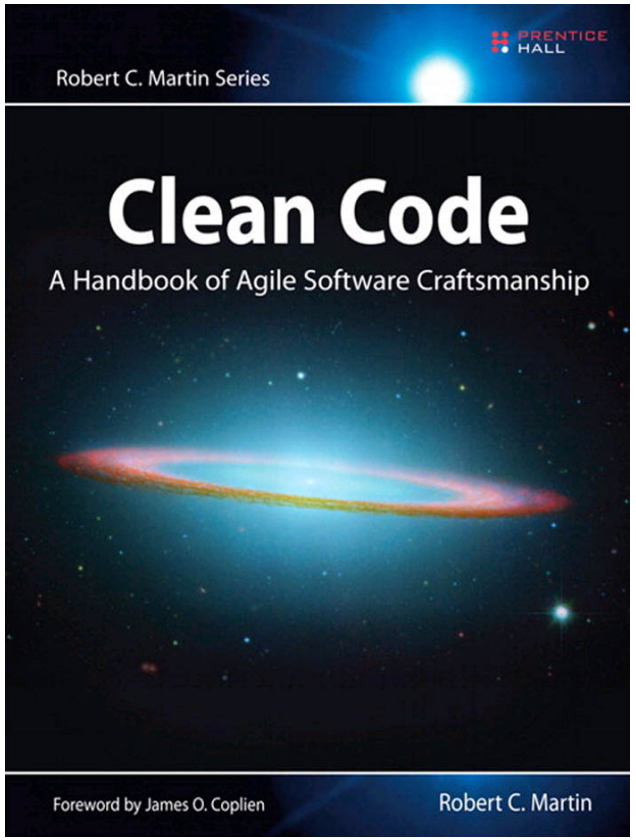
Implementation

- Röda tråden
- Software Craftmanship
- **Clean code**
- Code Quiz
- Teststrategier
- Vad är en bra implementation?

Clean Code

What is it?

Some excerpts from:



Clean Code?

```
<!-- left_navigation_eof //-->
    </table></td>
<!-- body_text //-->
    <td width="100%" valign="top"><table border="0" width="100%" cellspacing="0" cellpadding="0">
      <tr>
        <td><table border="0" width="100%" cellspacing="0" cellpadding="0">
          <tr>
            <td class="pageHeading"><?php echo HEADING_TITLE; ?></td>
            <td class="pageHeading" align="right"><?php echo tep_image(DIR_WS_ICONS . 'table_background_history.gif', HEADING_TITLE, HEADING_IMAGE_WIDTH, HEADING_I
          </tr>
        </table></td>
      </tr>
      <tr>
        <td><?php echo tep_draw_separator('pixel_trans.gif', '100%', '10'); ?></td>
      </tr>
      <tr>
        <td>
<?php
  $orders_total = tep_count_customer_orders();

  if ($orders_total > 0) {
    $history_query_raw ="select o.orders_id, o.date_purchased, o.delivery_name, o.billing_name, ot.text as order_total, s.orders_status_name, s.visible_for_custome
                         "from " . TABLE_ORDERS . " o, " . TABLE_ORDERS_TOTAL . " ot, " . TABLE_ORDERS_STATUS . " s ".
                         "where o.customers_id = '" . (int)$customer_id . "' and o.orders_id = ot.orders_id and ot.class = 'ot_total' and "."
                         o.orders_status = s.orders_status_id and s.language_id = '" . (int)$languages_id . "' and ".
                         "not exists (select 1 from dkny.orders_status_history t where t.orders_id = o.orders_id and t.orders_status_id =".ABORTED_PAYMENT_STATUS.")
                         "order by o.orders_id DESC";
    $history_split = new splitPageResults($history_query_raw, MAX_DISPLAY_ORDER_HISTORY);
    $history_query = tep_db_query($history_split->sql_query);

    while ($history = tep_db_fetch_array($history_query)) {
      $products_query = tep_db_query("select count(*) as count from " . TABLE_ORDERS_PRODUCTS . " where orders_id = '" . (int)$history['orders_id'] . "'");
      $products = tep_db_fetch_array($products_query);

      if (tep_not_null($history['delivery_name'])) {
        $order_type = TEXT_ORDER_SHIPPED_TO;
        $order_name = $history['delivery_name'];
      } else {
        $order_type = TEXT_ORDER_BILLED_TO;
        $order_name = $history['billing_name'];
      }

?>
          <table border="0" width="100%" cellspacing="0" cellpadding="2">
            <tr>
              <td class="main"><?php echo '<b>' . TEXT_ORDER_NUMBER . '</b> ' . $history['orders_id']; ?></td>
              <td class="main" align="right"><?php
                if($history["visible_for_customer"] == 0)
                {
                    $statuses_query = tep_db_query(
                        "SELECT os.orders_status_name, osh.date_added, osh.comments ".
                        "FROM " . TABLE_ORDERS_STATUS . " os, " . TABLE_ORDERS_STATUS_HISTORY . " osh ".
                        "WHERE osh.orders_id = '" . $history['orders_id'] . "' and os.visible_for_customer = 1 and osh.orders_status_id = os.orders_status_id and o
                            "' order by osh.orders_status_history_id desc");
                    $tmp = tep_db_fetch_array($statuses_query);
                    $history['orders_status_name'] = $tmp['orders_status_name'];
                }
```

Clean Code

Be proud!

Cleverness in code is never a thing to be proud of.

Four principles of simple design by Kent Beck:

(1) Tests pass

(2) Exposed intent (good names)

(3) DRY

(4) Small.

Clean Code - Hur?

Ubiquituous Language
Design principles
Coding guidelines
Continuous Design – **refactor**!
TDD!
Patterns
Pair programming

Coding guidelines

Size – must fit on screen!
Naming
Indentation
Commenting
File naming
Etc…

Size

Method must fit on screen

Class must fit on screen

Refactor till you drop!

Method not more than 5 lines

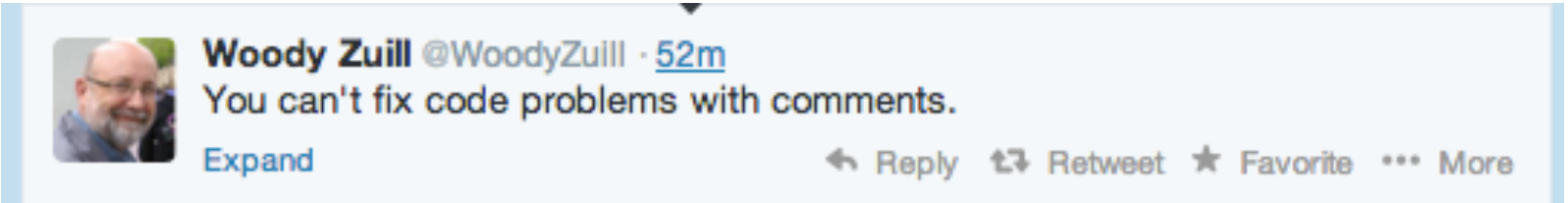
Naming

Comments

Comments is a smell

Be proud!

Comments are never a thing to be proud of.



Comments does not compile!

Never

- Repeat Code
- Explain line
- Keep TODO, FIXME, etc

OK

- External refs
- Summary
- Code intent

Refactor

Bad naming
Duplication
Long method/Class
Bad Design
    Not SRP
    Not DRY
    etc…
Slow
etc…

Rename
Extract method
Extract class
Move
Substitute Algorithm
…

Boy Scout Rule

Broken Windows

Code Smell

- "a code smell is a surface indication that usually corresponds to a deeper problem in the system” - Martin Fowler

- Circular Dependencies

- objectA.getObjectB().getObjectC().doSomething();

Teknisk skuld

Metafor skapad av Ward Cunningham (Wiki, XP, Fit, design patterns, …)

Två lösningar

1. Snabb, men kommer att göra förändring svårare längre fram
2. Tar längre tid, men renare design och tydligare kod.

Lösning 1 skapar en skuld, som måste betalas av.
Om man inte betalar av så blir räntan till slut för hög.

Teknisk skuld

Saknade tester är INTE teknisk skuld i ursprungsmeningen!
Det är bara dumt!

Implementation

- Röda tråden
- Software Craftmanship
- Clean code
- **Code Quiz**
- Teststrategier
- Vad är en bra implementation?

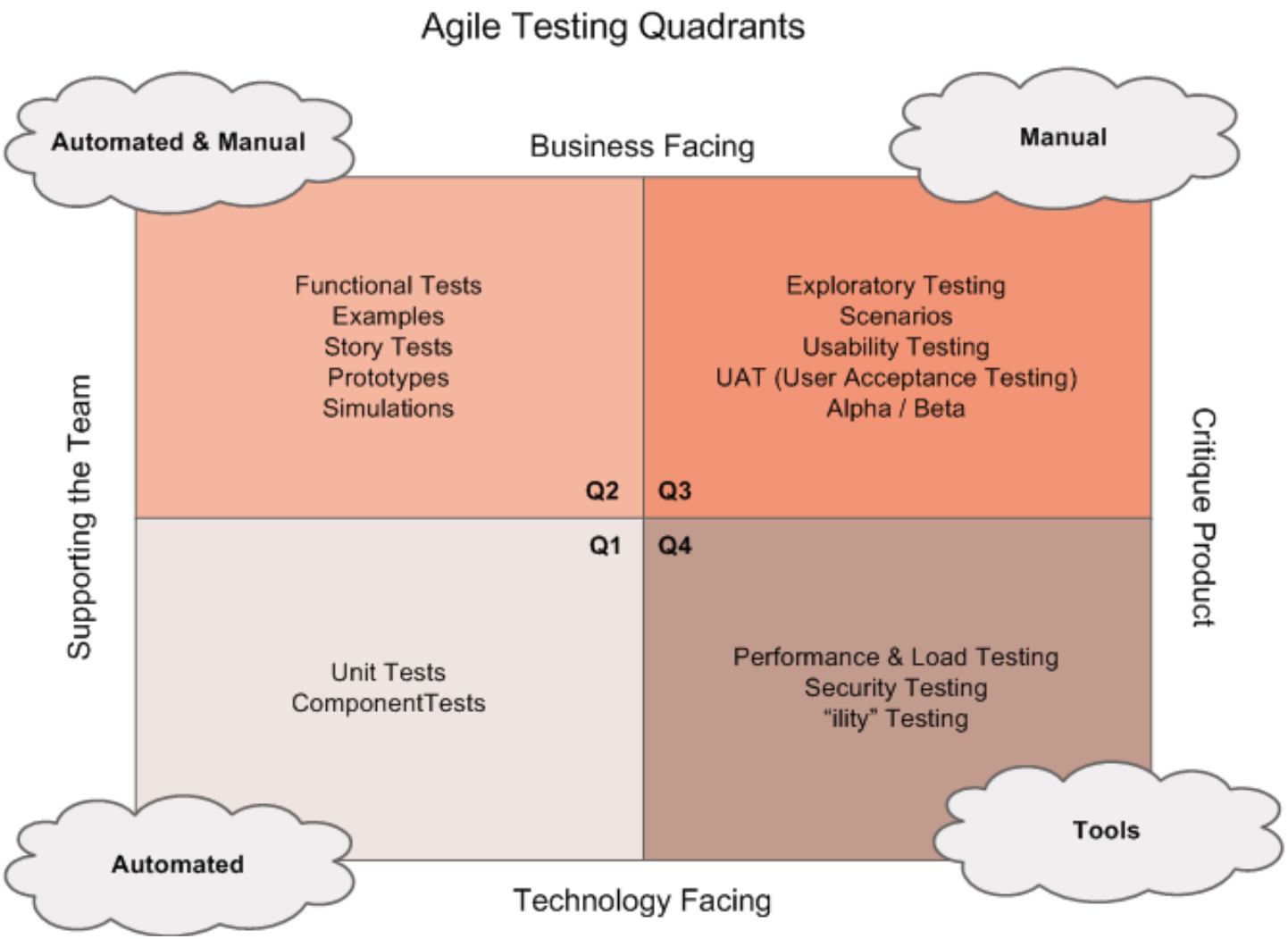
Code Quiz!

Hittas på https://github.com/jonananas/teaching

Implementation

- Röda tråden
- Software Craftmanship
- Clean code
- Code Quiz
- **Teststrategier**
- Vad är en bra implementation?

Testing



http://lisacrispin.com/2011/11/08/using-the-agile-testing-quadrants/

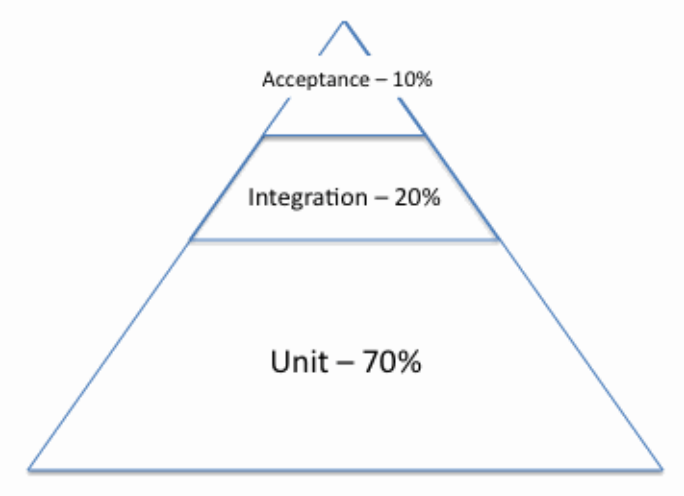
Test Automation Pyramid

http://www.minecraftforum.net/forums/mapping-and-modding/maps/1503376-insanely-enormous-multiplayer-puzzle-pyramid-

Test Automation Pyramid

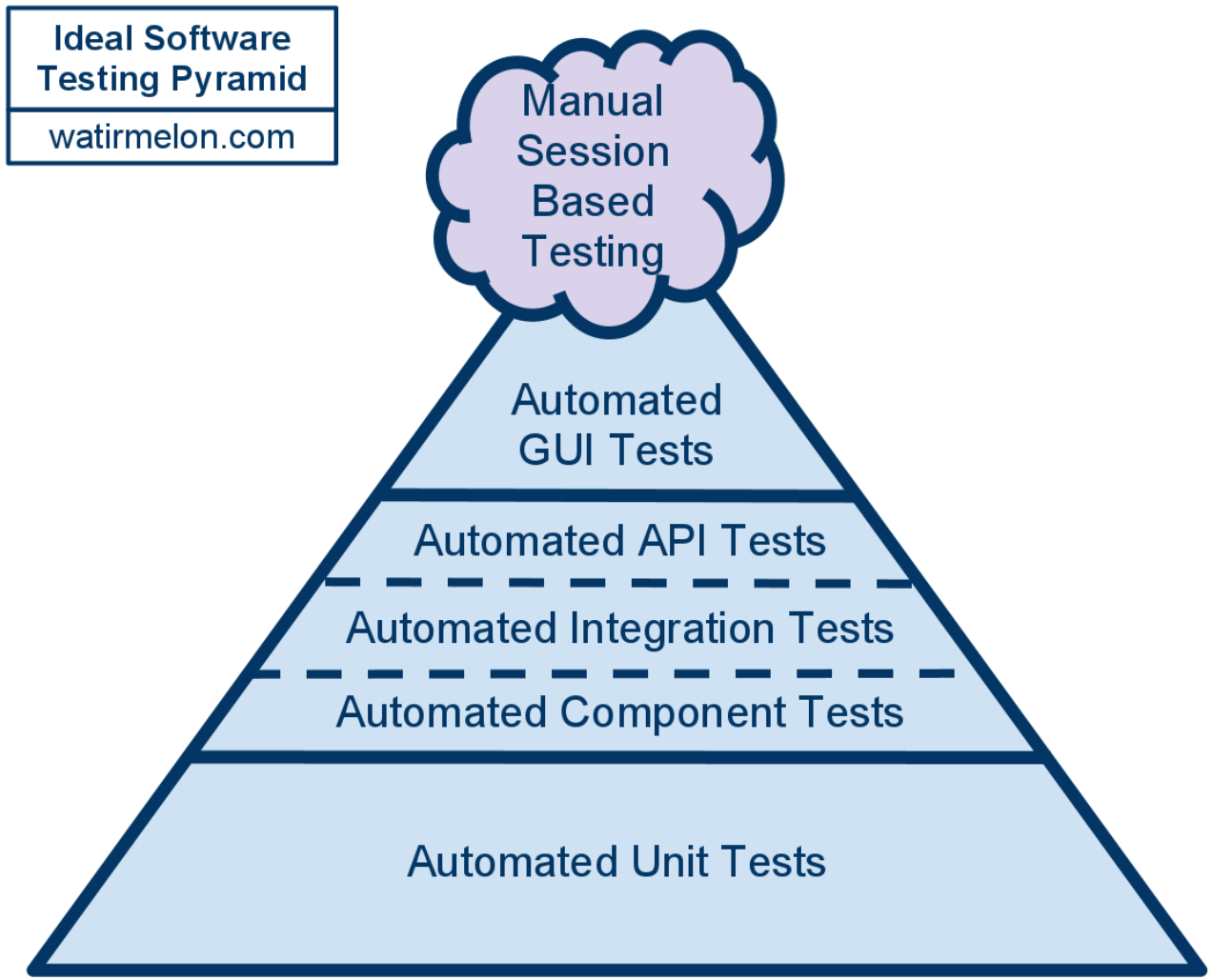
Mike Cohn

http://www.mountaingoatsoftware.com/blog/the-forgotten-layer-of-the-test-automation-pyramid



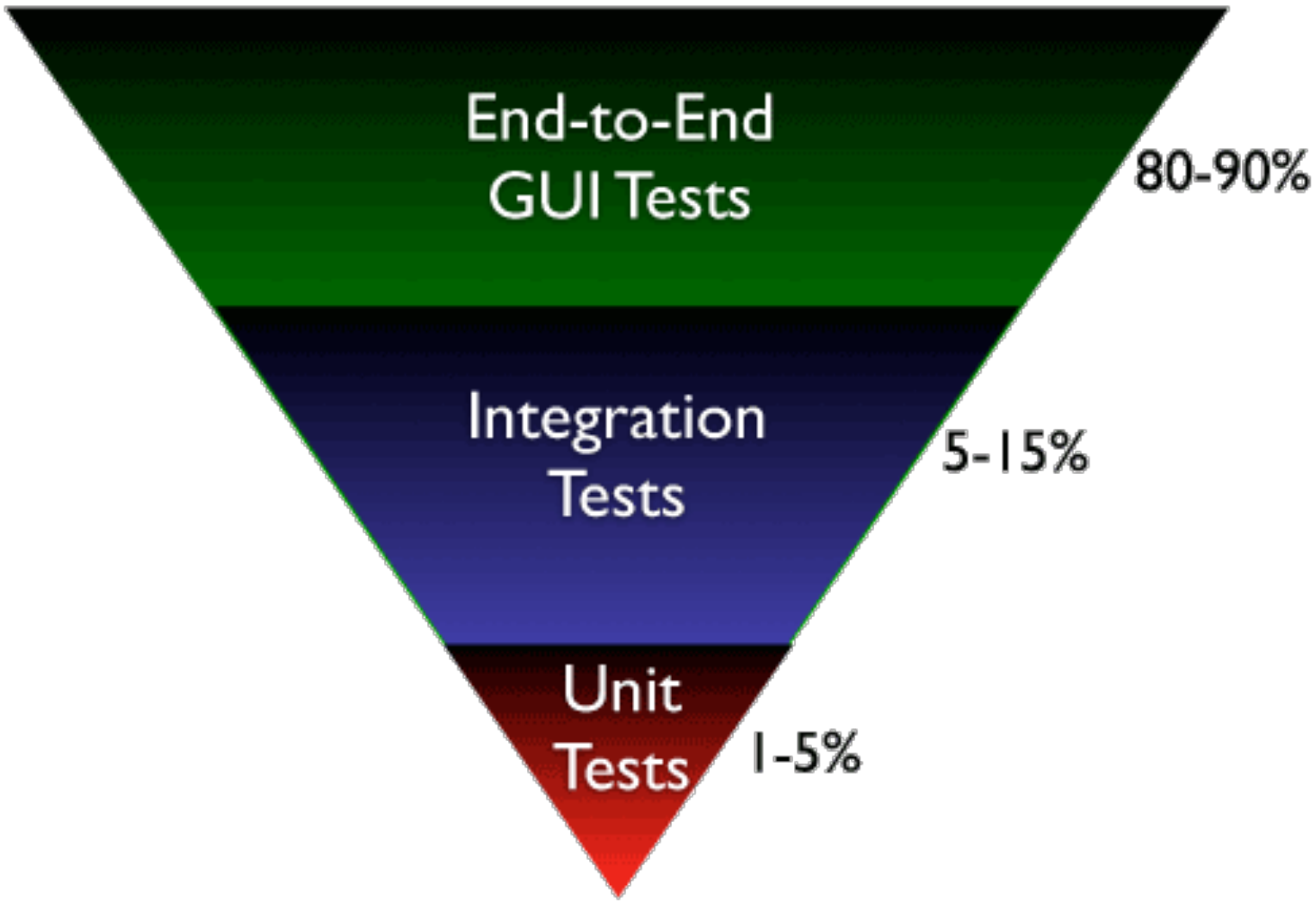
http://jamescrisp.org/2011/05/30/automated-testing-and-the-test-pyramid/

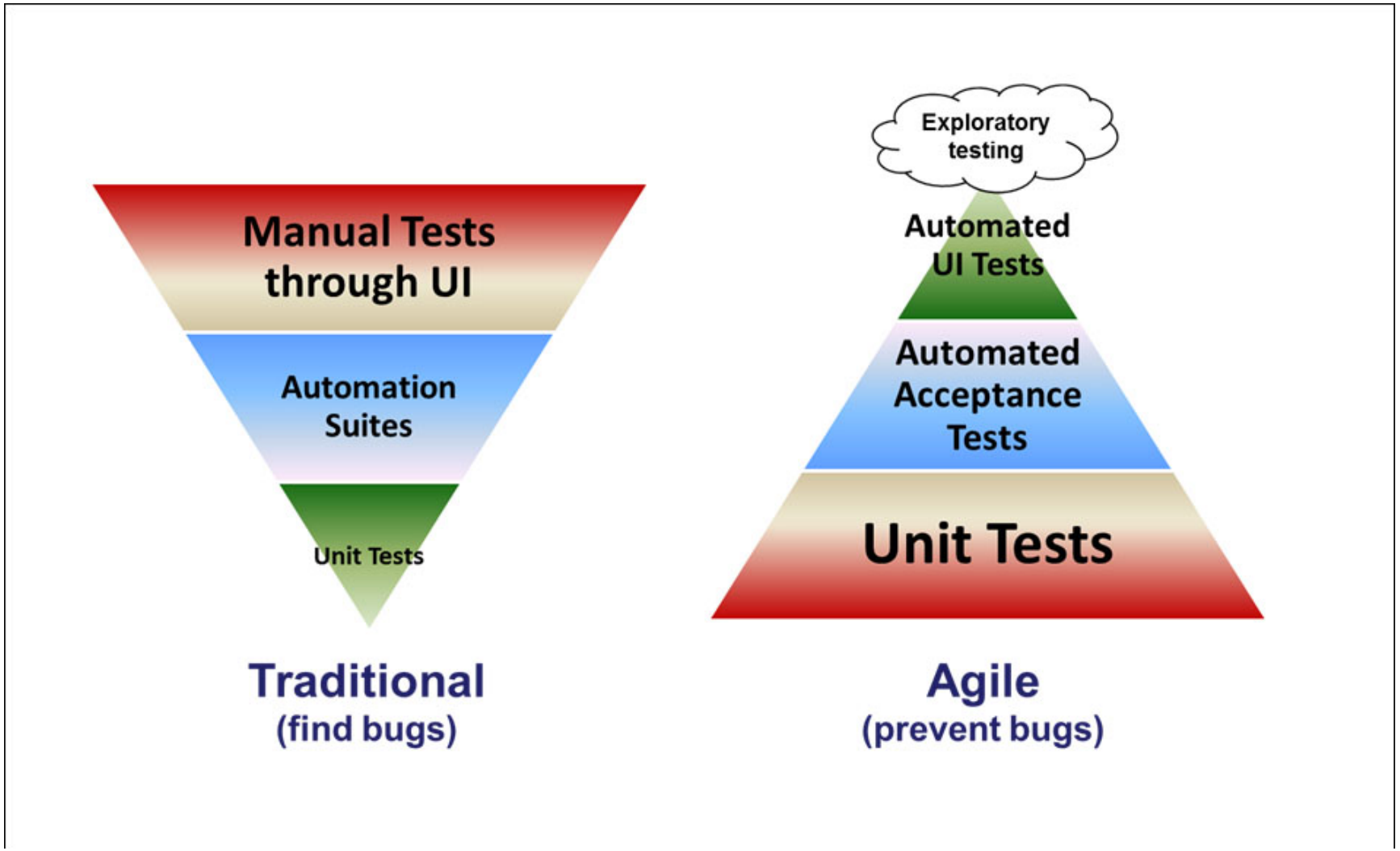
Test Automation Pyramid



http://watirmelon.com/tag/testing-pyramid/

Ett annat sätt...





http://www.agilecoachjournal.com/index.php/2014-01-28/testing-2/the-agile-testing-pyramid/attachment/agiletestingpyramid-3/

Testing

- Acceptance Testing
- UI Testing
- Conformance Testing
- Configuration Testing
- Performance Testing
- Stress Testing

Implementation

- Röda tråden
- Software Craftmanship
- Clean code
- Code Quiz
- Teststrategier
- **Vad är en bra implementation?**

Good implementation?

Good implementation - ilities

Completeness
Correctness
Traceability
Performance
Readability
Maintainability

Good implementation

Complete
Correct
Traceable
Performant
Readable
Maintainable

Good enough!

Complete *enough*
Correct *enough*
Traceable *enough*
Performant *enough*
Fast *enough*
Readable *enough*
Maintainable *enough*

→ **Customer Value**
(Cheap enough)

Complete and Correct

Completeness – gör **allt** i kravspec
Correctness – löst enligt kravspec

Några problem med det?

Det beror på! (ständiga konsultsvaret)

Levande specifikation → Ja!

Up-Front-Specification → Nej!

Complete and Correct

Målet med BDD/ATDD/Specification By Example

En kontinuerligt exekverande kravspec!

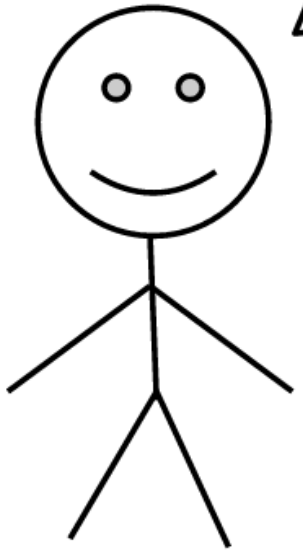
Traceability

Process och verktyg!

T.ex. Wiki + Jira + Subversion

Performance

We should forget about small efficiencies, say about 97% of the time: **premature optimization is the root of all evil**. Yet we should not pass up our opportunities in that critical 3%



Donald Knuth

Det innebär dock inte att prestanda kan testas sent!

Performance

Programming Wisdom @CodeWisdom

"The cleaner and nicer the program, the faster it's going to run. And if it doesn't, it'll be easy to make it fast." - Joshua Bloch

Readability
Maintainability

Lätt att
- Förstå
- Ändra → Clean!
- Använda
- Testa
- Integrera

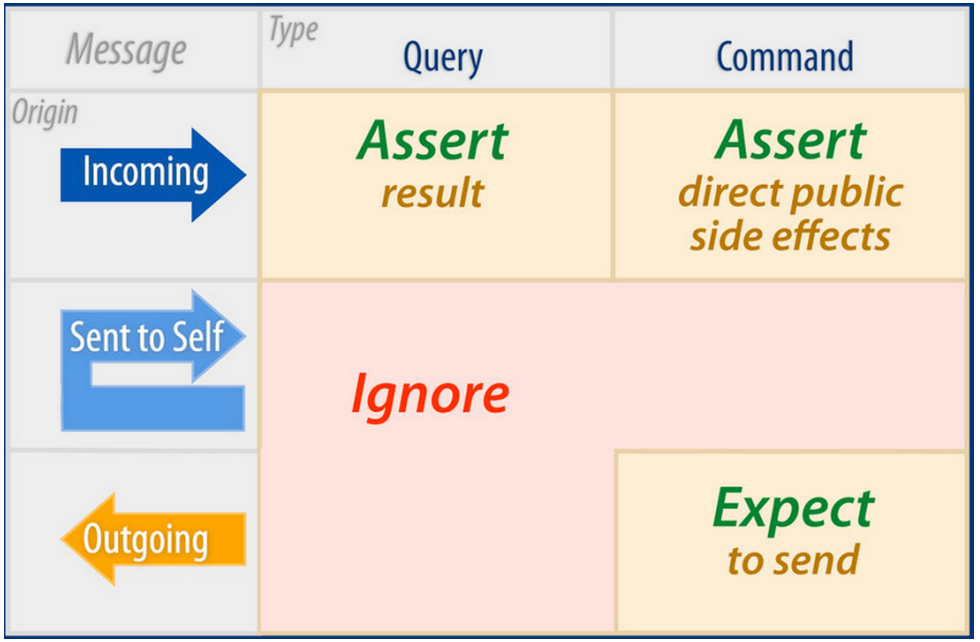
Good implementation

Complete
Correct
Traceable
Performant
Readable
Maintainable

Film inför 24/4

The Magic Tricks of Testing by Sandi Metz 32 min
https://www.youtube.com/watch?v=URSWYvyc42M

För tips om hur man undviker sköra tester



Imorgon

Automated builds (maven)

- Installera maven i förväg!
- git pull https://github.com/jonananas/pvt
- Fundera på fler saker ni vill ta upp!