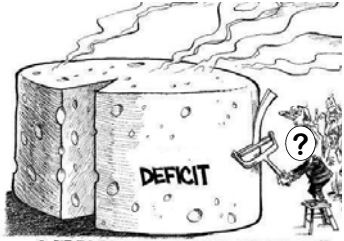


Regression Deficit: The Danger of Waiting to Test



David Hussman – DevJam

© 2008 DevJam - All rights reserved.

Too Common A Disorder

Regression Deficit Disorder (RDD)

- Features / stories are coded
- Code is committed
- Tests are not run
- Work is deemed “done”
- Issues are not found (or found late)
- Test costs and issues grow unchecked

© 2008 DevJam - All rights reserved.

Who Is Suffers from RDD?

Struggling Projects

Poor quality
Unhappy users
Declining market share

Successful Products

Huge demand
High volumes

© 2008 DevJam - All rights reserved.

How Much Do You Test?

Number automated tests?

Percent code coverage?

Use of feedback from tests?

Ratio of tests to work completed?

How Much Is Enough?

© 2008 DevJam - All rights reserved.

Deficit Inducers

Reasons Not To Automate

1. "We do not have enough time"
2. "Our problem is too hard to test"
3. "You cannot automate our problem"
4. "We do not know where to start"
5. "My people do not see the value"

© 2008 DevJam - All rights reserved.

1) “We Don’t Have Time”

How much time is available?

Too little time or too much to do?

“The tests take too long to run”

*“We can simply off shore the problem”
(... to save time?)*

© 2008 DevJam - All rights reserved.

2) “Problem Is Too Hard”

Optical Research Associates

Tool ignorance or lack of tool innovation

All or nothing attitude
(no incremental investments)

© 2008 DevJam - All rights reserved.

3) Automation Not Possible

Blocked by bad tools

Overusing one tool

Garbage in, garbage out
(sick environments are groove killers)

Data blockers

© 2008 DevJam - All rights reserved.

4) Where Do We Start?

Determine most valuable type of tests

Next best tests provides most feedback

One test at a time – Start somewhere!
(small, incremental investments)

© 2008 DevJam - All rights reserved.

5) “Show Me The Money” (where is the value?)

Success sells

The best incentive is experience

Testing doesn’t happen because
people are told to do so

Visible feedback feeds test infection

© 2008 DevJam - All rights reserved.

Available Antidotes

And

Anecdotes

What Is Your Blend?

Who?
(people)

How?
(process)

What?
(tools)

© 2008 DevJam - All rights reserved.

People

Some people love to test
(personal motivation)

Everyone owns quality
(environmental support)

The days of "I just code" are over
(job security)

© 2008 DevJam - All rights reserved.

Process

Short cycles

Meaningful feedback

Shared definition of "done" or quality

*There needs to be time and a place
to invest in automated testing ...*

© 2008 DevJam - All rights reserved.

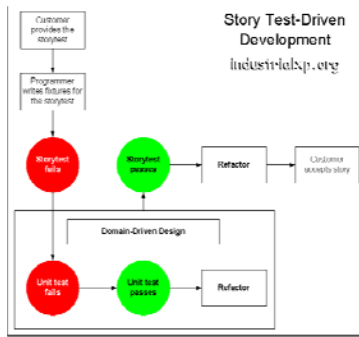
The Gathering



Tests Define Done

© 2008 DevJam - All rights reserved.

An Innovative Process



© 2008 DevJam - All rights reserved.

Tools

Micro (code) Tests: xUnit – TestNG

the story of the code

Macro (functional) Tests: FIT – Selenium

the story of the application / system

Regression Tests: Any of the Above

the health of the system

© 2008 DevJam - All rights reserved.

Metric Mashups

Work completed verses test coverage

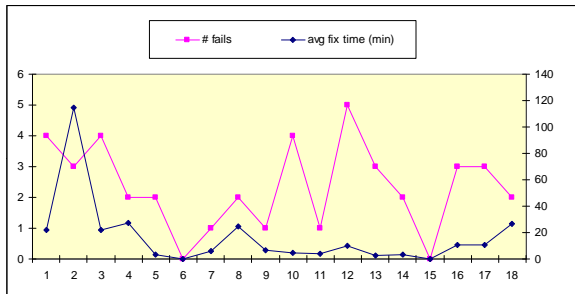
Frequency of test runs verses issues found

Time to fix the build (with tests included)

Any comparison that radiates health!

© 2008 DevJam - All rights reserved.

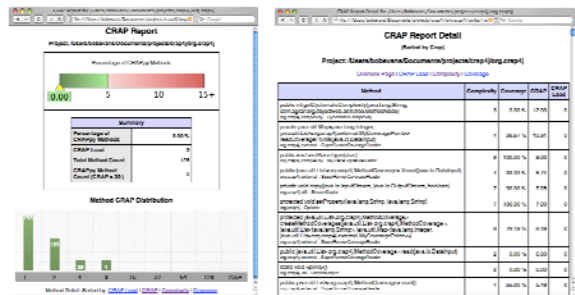
Health of the Build



What is a broken build?

© 2008 DevJam - All rights reserved.

Change Risk Analysis & Predictions



C.R.A.P. Code (www.crap4j.org)

© 2008 DevJam - All rights reserved.

What will stop you?

OR

What will help you?

Questions To Ponder

Who do you hire to test?

Is your testing focus too limited?

Are your testing tools broken?

Are your testing investments wise?

© 2008 DevJam - All rights reserved.

Ping Me, I Will Ack Back!

david.hussman@devjam.biz

www.devjam.biz

***New Book - "Cutting An Agile Groove"
Pragmatic Series - Due out Early 2008***

© 2008 DevJam - All rights reserved.
