Table of Contents

Premise

In this talk we'll look at why the technical practices of test-driven development, refactoring, continuous design, clean code and automated testing can help you and your organization be great. This talk is not just for the technical people. Business people need to understand that they cannot have a great product and productive team without technical excellence.

Technical excellence is more than two week sprints, a burn-down chart and a daily stand-up meeting. The basic rules of Agile or Scrum are not an end in themselves, but rather a staring point based upon principles and practices that allow and encourage teams to adopt, adapt, and refine their craft. Unfortunately, it too often seems that agile is just another micro-management approach.

Extreme Programming, the spur under the saddle that started this wild ride, is based on sound technical practices. Why do so few employ the engineering practices that are designed to support the tight iterative cycles of Agile and Scrum? The founders of Scrum expected you to pull in the engineering practices. They figured that once the continuous improvement cycle revealed the problems of poor product quality, hard to change code, wasted time debugging, long stabilization efforts and the ever growing burden of manual test, you'd hunt for solutions. Come to this session and see why you can't be great without technical excellence.

Summary

- Content rating (0-no new ideas, 5 - a new ideas/approach, 9-new ideas): 6 some new ways of thinking about things
- Style rating (0-average presentstion, 5 - my level, 9-I learned something about presenting): 4 bitty

Action / Learning

Presentation

Notes

Book "the debrief imperative" Murphy

Scrum, instead of PDCA people just to A

Agile 10 year anniversary in Snobird

1. Demand technical excellence
2. Promote individual change and lead organizational change

Symptom is hardening sprint Does this degrade into a firefight

Being good at chasing bugs does not mean technical excellence

Propose a marriage of scrum and XP

Certified scrum developers 55K vs 300K certified scrum masters

How are you spend your time mr developer Coding, testing, debugging

Debug later programming vs tdd

Longer time it takes discover a bug, increases time to find it in the code. Time to fix is short.

We create the bugs. We did it.

We have a problem

People do what they know I've got 10 years experience - the same year 10 times doesn't count

XP

Talks about values - simplicity, respect, feedback Practices help Dijkstra - not waste time debugging, should not put defect in

If we move test upstream could we get working features to flow

Tdd is not so much test first and test together

Changes the physics of the mistake.

A complex system that works has evolved from a simple system that works

Book systems bible - John Gall

Why not do automated test at module level. 3 modules means all tests would take 1000 tests Unit test strategy 10 units test per module plus a few of of the interactions

Orgs are set up to assuming amount to test is same to create - we know as fixed resources

25% defects introduced while changing and fixing code. Means 75% were introduced new code

Because systems act up we have to be careful

Cannot just what was done in previous iteration, as have to test what we did in the past Unsustainable

growth in the untested code gap

Tests must be automated Cost of retest is low

Tests is not all there technical excellence

Two values of software Want to be able to evolve our code

Refactoring 3 critical skills Smell bad code and recognize why it is bad Step wise refinement of code. Requires tests.

The lawyers are coming Toyota brake problem was caused by single bit flip in the code

Why doesn't team use tdd and refactoring

Won't work for us

Stories are optional until we are late You are all special But it doesn't matter

Find a way to get the product shippable

It is hard

Don't have time compared to what?

Michael Dubakov looking at optimize development speed Green help. Red detracts. Yellow could have effect either way

No wonder we have any time

Tdd is faster than debug later programming

Fun - working proactively, getting things done

Scrum masters Make sure your team knows about these state of the art practices Support learning, problem solving and continous improvement Make sure org understand what is important Not dogma followers

Developers Become expert in your craft Build trust - make your work visible Learn and try things Become advisor to org Become aware of limits of org Pair

Managers Stop motivating people

Motivating people is hard, demotivating is easy

A programmers life Work, home, play, sleep - we always think about code. Title: JamesTamm-WantBetterCollaborationDontBeSoDefensive Timestamp: 2015-08-07 16:02:46 +0000 Created: 2015-08-07 11:16:55 +0000 Last Accessed: 2015-08-07 11:16:55 +0000 Times Accessed: 0 Tags: Metadata: gpslongitude=-77.017185,gpslatitude=38.781725 # Premise

Jim's keynote will challenge you to look at yourself through self-critical eyes. Entitled "Want Better

Collaboration? Don't be so Defensive!", Jim will discuss skills essential for effective collaboration. In particular, he will focus on achieving success even during difficult interactions. He will show how your own defensiveness is a key factor in resolving conflicts and building collaboration. He will share practical tools designed to help you manage your own defensiveness.

For most of his career, Jim was a Senior Administrative Law Judge for the State of California. He had jurisdiction over public sector disputes in the workplace. He mediated over 1,500 labor disputes, including more school district labor strikes than any other person in the United States. For several years he was a member of a collaboration special task force. They designed and taught collaboration skills to highly conflicted public sector organizations. The project was wildly successful. It helped build trust, reduce conflict and create more collaborative working environments.

Jim is President of RC Group, LLC. He maintains offices in South San Francisco, California, Cuernavaca, Mexico and Stockholm, Sweden. He specializes in building cultures of collaboration within organizations. He also trains other consultants and trainers how to teach collaborative skills.

Summary

- Content rating (0-no new ideas, 5 - a new ideas/approach, 9-new ideas):
- Style rating (0-average presentstion, 5 - my level, 9-I learned something about presenting):

Action / Learning

- watch - Ted talk

Presentation

Notes

Chicken story Breeding 1 year aggressive 260% increase in eg laying, 55% less mortality

Red zone - produce more red zone behavior Green zone - same for green More eggs and lower mortality

Can be applied to humans

See stats for green and red companies over 11 years Kotter and hasket

Cannot compete externally if you are competing internally

5 key skills to improve collaboration Mindset and practice

Collaborative intention Stay focused on mutual gains in you relationship Stay in green zone and get curious or go in the red zone and get furious

Are you operating in the red zone on the green zone? Start in red zone Just by paying attention every day got to green zone

Notice if you are in the red zone / green zone

What pushes you into the red zone? Pulls helps you return to green zone?

Truthfulness Safe enough to raise difficult issues

Death by 1000 cuts is way it manifests itself

Self accountability

What choices are available What choices are being made

Not - the made us do it

Be accountable for unintended and intend consequences

Self awareness

Build awareness about your own self defensive

Negotiating and problem solving

System of negotiation would not help if you arrive at table with wrong attitude

Could help with individuals as well

Biggest killer to collaboration Better managing own defensiveness

Defensiveness budget More meetings Revisiting Triangulating People not tell the truth

Think about defensiveness. What does it cost.

If reduce defense you increase ability to solve problems

Therefore stay in green zone

Become one with the piece of paper Deep breath in out Scrunch up (lead) - anyone want to collaborate with me See how quick it changes

Defensiveness does not protect us from other people It defends use from fears we don't want to feel

Big areas My competency

I might start blaming

Putting whipped cream on dog poo Look between but doesn't deal with underlying issue

Not always aware of our fears therefore look at the defensive behaviors

What are your defensiveness See list Check which ones apply to me Circle top 3

Share with neighbor Not secret information

This is a human condition

Useful also to know what the defensive behaviors of colleagues

Personalized warning system To stop you becoming less effective

What do about it Acknowledge that you are getting defensive (perhaps let others know as well) Slow down (your phisoloogy - eg go for walk, visualize calm place) Check negative self talk - turn into something a little less toxic for yourself Create a personalized action plan Start over

Start noticing if in red zone / green zone Look for defensiveness Practice your action plan