Crafting Code



January 19, 2018 Harald Störrle <u>hstorrle@acm.org</u> @stoerrle <u>https://www.pst.ifi.lmu.de/~stoerrle/index.html</u>

GAME OF CODE A TALE OF QUALITY AND AGILITY

" Considering the current sad state of our computer programs, software development is clearly still a black art, and cannot yet be called an engineering discipline."

Bill Clinton

The Gartner Hype-Cycle



Hawthorne works, PA, ca. 1930





Anecdotes are not scientific evidence



The highly respected Professor Nibbowitz proved, that octopus are more intelligent than cat, when exposed to the same challenges and conditions.





...this probably more realistic.

0

9

0

0.10

0.0

1007 F R

TUU

Quality is free – or is it?

"Quality is free. It's not a gift, but it is free. What costs money are the unquality things – all the actions that involve not doing jobs right the first time."

Philip B. Crosby "Quality is free"

"Quality is free, but only to those who are willing to pay heavily for it."

T. DeMarco and T. Lister "Peopleware"

(Formal) Inspections

- Code inspection pays for itself during development, isolates defects much more efficiently than testing, and has been shown to reduce maintenance costs by 90%.
 - Quality processes reduce errors without increasing costs.
 Increased quality assurance is associated with a decreased error rate but does not increase overall development cost.
 - Inspections cut rework by half.

Raytheon reduced its cost of rework from about 40% of total project cost to 20% through an initiative that focused on inspections.

Inspected software costs 90% less to maintain.

ICI found that maintaining a portfolio of about 400 programs was only about 10% of the cost of maintaining a similar set of programs that had not been inspected.

Inspection catches 70% of defects.

The combination of design and code inspections usually removes at least 70% of the defects in a product.

Inspection catches 60% more errors than testing.

Code inspection at NASA's Software Engineering Laboratory found 20 to 60 percent more errors than testing did.

- 85% of errors are trivial.

About 85% of errors can be fixed in a few hours.

Static Analysis

• Powerful static analysis techniques are implemented in many tools and environments.

- They are easy to use.

In many cases, it is no more complex than to run your tests on the Jenkins server. Interpretation of the measurements may require some brain capacity

Tools are readily available, and cheap.

There are several great, well-integrated, tools: SonarQube is LGPL, Structure 101 is commercial but affordable. Training and consulting are available at reasonable cost and it's money spent wisely.

- Static Analysis can be essential in competitive situations.

If you can guarantee measureable quality levels, and your competitors cannot, you have an advantage, and may even be able to demand higher prices. By offering defined quality levels to your customers, they may develop a taste and demand it from your competitors as well.

- Static analysis is not complete.

The combination of design and code inspections usually removes at least 70% of the defects in a product.

Many simple defects are found earlier, cheaper, and with less embarrassement.

Don't wait until your client/boss finds out about your code quality.

Beware of Sonar-blindness, though.

Not all artifacts are covered (configuration data, some languages), and not all warnings are really problems. Blind trust in measurements can be counterproductive.

Technical Debt

- Technical Debt is a metaphor for short-term goal satisfaction at the expense of long-term goals.
 - Apart from poor code, this also covers poor documentation, communication, relationships etc.
- Taking out a loan comes back at you, with interest, depreciation, and technical, inflation.
 - At the least, you should be aware that you are indebted.
- Taking out a loan can be a good business decision, if it is informed, and controlled.
 - If not, it is a reckless gamble.
- Best Practices, standards, authorities' opinions, your bosses decision, or the tool's output are no replacement for critical thinking and common sense.
 - All of these may inform your decision, but it is your decision all the same.
 - And your responsibility, in the end.



TODAY: USE LAYERS TO DECOUPLE



ANNUAL RINGS

[http://geekandpoke.typepad.com/geekandpoke/2011/03/architectural-best-practices.html]





Practical Advice

- **1.** Use static analysis tools
- 2. Use formal (!) inspections
- 3. Measure, track, and improve
- 4. Do post-mortems
- 5. Document what you do
- 6. No code without tests
- 7. Demand evidence for claims
- 8. Demand reasonable and argued goals from leaders/clients
- 9. Take responsibility for your work and be proud of it
- **10.** Be active in the computing community



Code KARMA

When you write good code, others will understand. When you write your code based on good code created by some other programmer, it will be good code, too.

If you encounter horrible code, make it better. Don't just walk by, and leave it to be fixed by someone else. For you might be that one, and it will happen at a most unfortunate time.

One defect may cause a production error costing millions, and might just be your fault that this defect is still there. So be good, and do good...

Software is eating the world

Marc Andreesen, The Wall Street Journal, August 20, 2011. https://a16z.com/2016/08/20/why-software-is-eating-the-world/



RANKING THE WORLD'S MOST VALUABLE BRANDS

In just the last 10 years or so, tech brands have taken over the list



		0 1 2	3		
900) 496	simple iPhone game app	1			
	Unix v 1.0	1			
	Win32/Simile virus	1			
	average iPhone app				
	Pacemaker				
	Photoshop v. 1.0				
	Camino				
	Quake 3 engine 3D Video game system				
	Space Shuttle				
	a million lines of code				
	1 million	-1			
	a million lines of code 18,000 pages of printed text				
	War And Peace x 14, or Ulysses x 25, or				
	The Catcher in The Rye x 63	2	FTE	10kLOC	1mLOC
	3D video game system				
	Bacteria	1	Create	0.51	50100
	Age of Empires online				
	Age of Empires on the		Maint./1vr	0.005 0.02	0.52
	CESM Climate Model National Center for Atmospheric Research	1	• •		
	F-22 Raptor fighter jet		Maint./10vr	0.05 0.2	520
-	Linux Kernel 2.2.0				
	Jurassic Park codebase source: Dennis Nedry				
	Hubble Space Telescope				
	Unreal engine 3 3D video game system				
	Windows 3.1 1992				
	Large Hadron Collider (root software)				
	US military drone (control software only)				

Photoshop C.S. 6 image editing software

a million lines of code short page of production War and Phase of a so-binance 20. or binance 20. or CryEngine 2 Bacteria

Agent Complexe collect Agent Empires collect CESRA Climate Model F223 Repare fighter jet Union Kernel 22.0 general Automatic Park collect

Unreal engine 3 Windows 3.3

United States of States of

Market Mark

Macilla Core constante en la c Boeing 787

man de

// -----

and and and and

، دیده دیده دیده دیده

Intait Quickbooks Android Android Windows NT 4.0

> Linux 3 Open Office

F-05 Fighter jet

icrosoft Office 200

Windows 2000 -

> Windows 7 H

Windows 11 Facebook Debian 5.0 codebase Mac OS X "Tiger" Car software

Moune*

Google



ABOUT ACM MEMBERSHIP PUBLICATIONS SPECIAL INTEREST GROUPS CONFERENCES CHAPTERS AWARDS EDUCATION PUBLIC POLICY GOVERNANCE

Association for Computing Machinery

Advancing Computing as a Science & Profession

We see a world where computing helps solve tomorrow's problems – where we use our knowledge and skills to advance the profession and make a positive impact.



New Executive Director for CSTA

The Computer Science Teachers Association (CSTA) has named Jake Baskin as its Executive Director, effective February 5, 2018. Baskin is the Director of State Government Affairs at Code.org. He is a longtime CSTA member and champion for computer science education. As a high school computer science teacher in the Chicago public school system, Baskin focused on increasing access to computing for underrepresented groups and more than doubled female enrollment in introductory CS classes.



PEOPLE OF ACM INTERVIEW Meet Martin Wicke

Martin Wicke is a software engineer working on Google AI. He is a key member of the team that developed TensorFlow, a scalable machine learning software package that Google released as an open source project. Wicke presented the ACM Learning Webinar, TensorFlow: A Framework for Scalable Machine Learning C. "I believe machine learning will become pretty ubiquitous, driven by the gains that we see wherever it is applied today."



Career Center

Q Search

ACM India Annual Event 2018, February 15-17, Nagpur ₪

The ACM India Annual Event is a flagship event whose aim is to discuss trends in science and technology, and to celebrate ACM's spirit and India's accomplishments in computing. The event is attended by ACM Turing Award winners, ACM Office Bearers, and researchers and IT professionals. Moshe Vardi and Martin E. Hellman will keynote. The CS Pathsala Workshop, ISIGCSE Workshop and IRISS are co-located ACM events.



CRA Tech and Jobs Summit



Researchers and Practitioners:



ACM CEO Search

Quality starts today



No use procrastinating – no use complaining.

BREAKING NEWS



The Model Observatory

Why do people model and how do they use their models? Are there any differences between different groups? Does it pay to model, and if so: when and why?

Help us answer these questions and more by answering a few questions - it takes less than 5 minutes!



http://tinyurl.com/MU-survey-2014

Recommended Reading





Fred Brooks The mythical man-month 1975

DeMarco & Lister Peopleware 1987



Philip B. Crosby *Quality is free* 1979

- Robert C. Seacord: **Secure Coding (SEI,** August 13, 2009) https://www.sei.cmu.edu/webinars/view_webinar.cfm?webinarid=18652&gawebinar=securecoding
- Entertaining and thought-provoking quotes on Software Engineering http://www.softwarequotes.com/

Abstract: Crafting software

- Like many other academic disciplines, Software Engineering is a practical craft. Unlike medicine, law, or economics, however, Software Engineering is a fairly recent arrival on the scene. Its development is still explosive, and the impact is only starting to manifest itself in the public eye. We as representatives of this craft have a very special responsibility towards the general public, but first and foremost, it is our responsibility to do our job as best as we can, by the state of the art.
- In this talk I want to explore what we every one of us can do to live up to this standard, despite pressures on us from all side: impatient clients, economically pressed bosses, and colleagues not all of whom are always helpful. The good news is that it can be done, and that the tools and techniques are widely available to those willing to try. The bad news is that far too often, people get away cheating, cutting corners, and not paying back their (technical) debt. It is not a new story, many of the classic texts have already pointed out the issues decades ago (e.g., Mythical Man-Month, Peopleware, Quality is Free). One might say it is a well known tale of agility and quality. You might also call it the game of code.



The Speaker

Harald Störrle strives to make software better, and he believes that empirical research is the only way to do that. His main interest are methodology and software process, modeling and requirements.

In his day job, he is a principal consultant with QAware GmbH, Munich. Previously, he was a professor of software engineering in Denmark, Germany, and Austria, and worked in several companies in the IT industry. He is a Senior Member of the ACM and elected member of the ACM Europe Council.

Academia Industry

