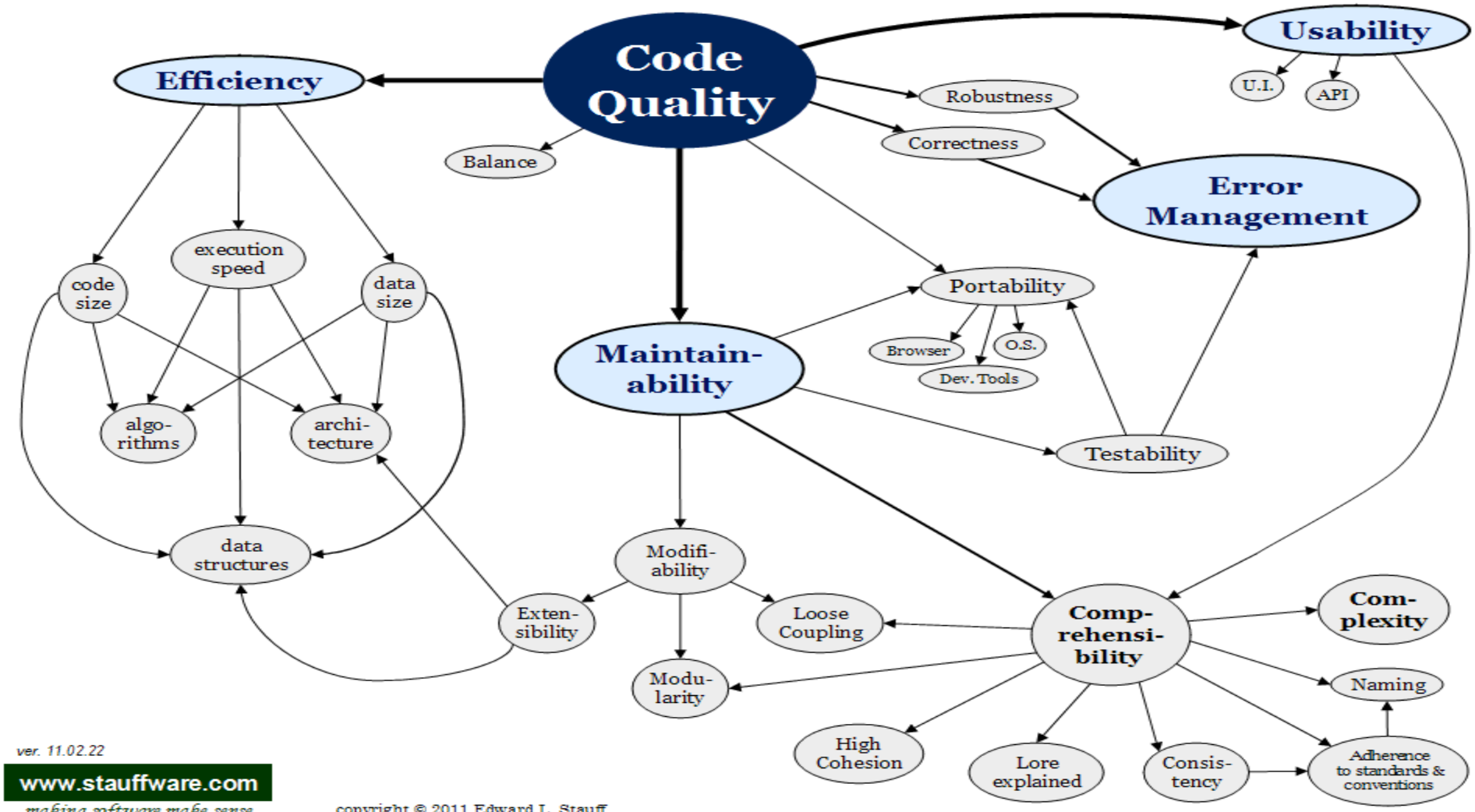
Tracking and improving software quality with

Ahmed Gomaa

Overview

- Code quality (what, why and when)
- The 7 Axes of quality and the Technical Debt
- SonarQube introduction
- SonarQube to the rescue (demo time)

What is Code Quality?



ver. 11.02.22

www.stauffware.com

making software make sense

copyright © 2011 Edward L. Stauff

What is code quality?

“It’s an indicator about how quickly developers can add business value to a software system”

Why measure?

- Source code is the heart of each system
- Developers don't write new software. They maintain "legacy" systems
- A system is (almost) never "finished"
- You can't improve if you don't measure
- Broken window theory

The broken windows theory

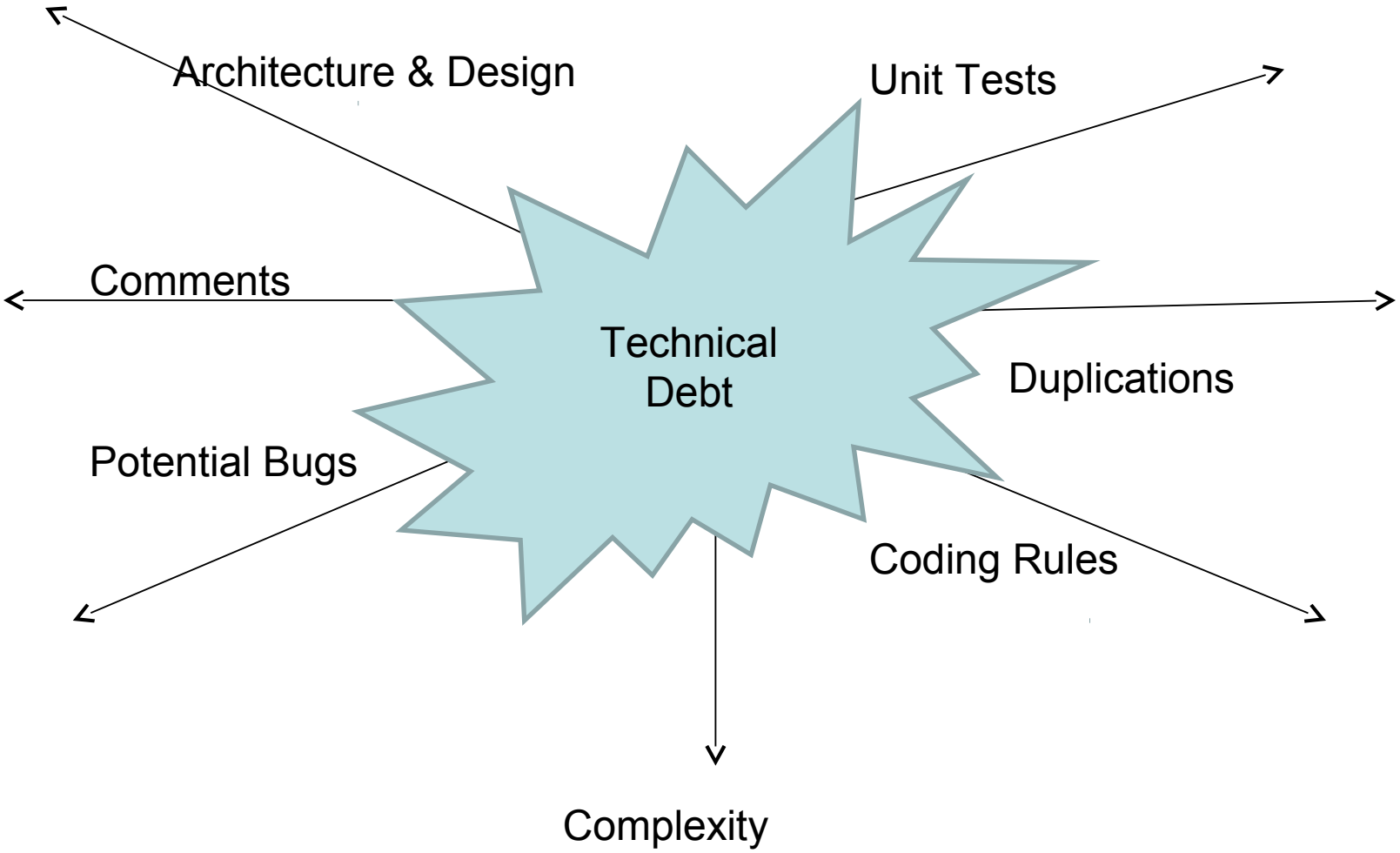
When you should measure?

- From project day #0
- Continuously
- Prevent vs post-actions
- Prioritize and plan

What you should measure?

- Not just abstract numbers
- Evolution through time
- Metrics?
- Welcome to the 7 deadly sins of devs

The Seven Axes of Quality



Technical Debt

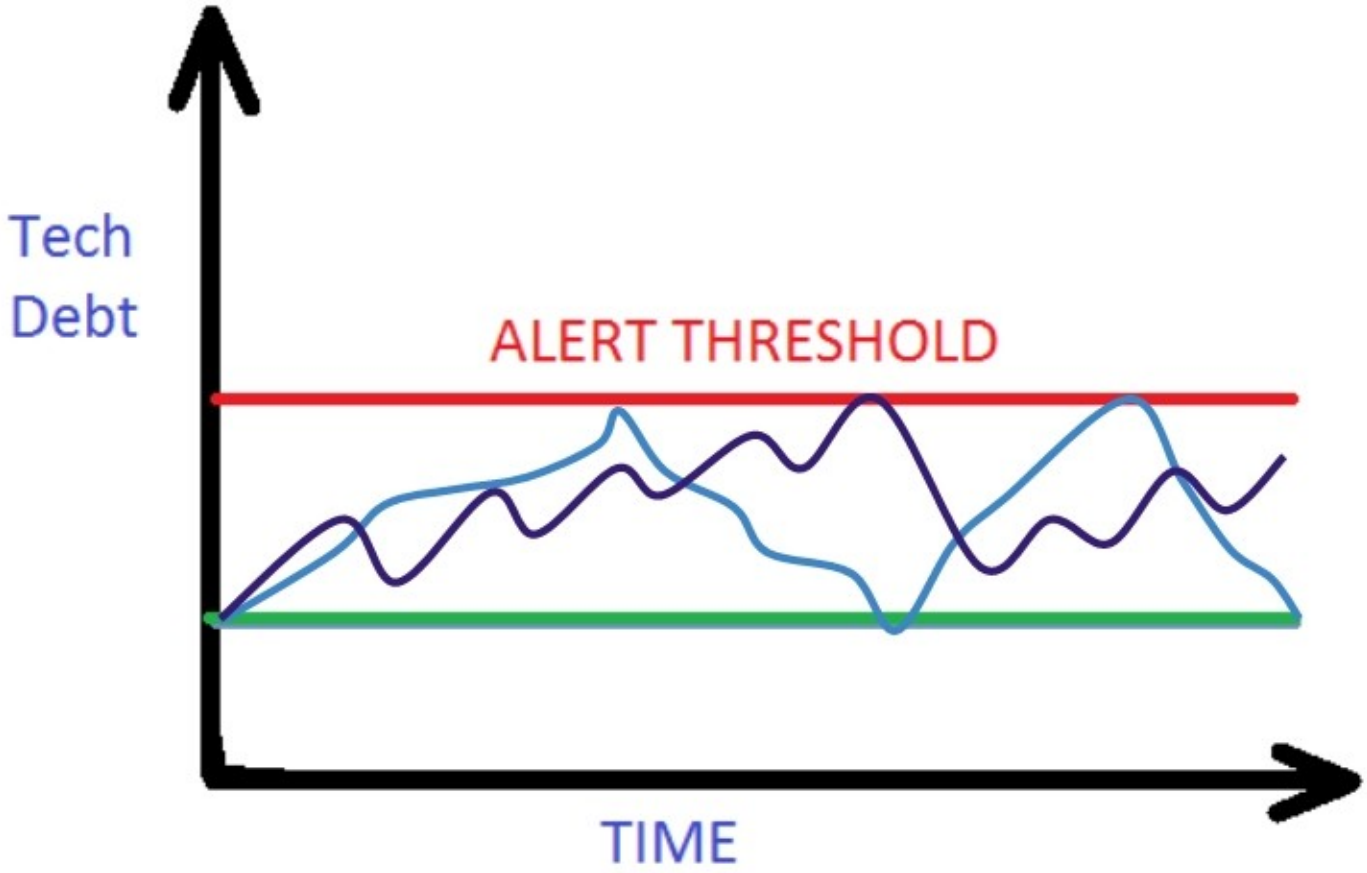
"If the debt grows large enough, eventually the company will spend more on servicing its debt than it invests in increasing the value of its other assets"

Steve McConnell

(author of code complete)


This is your source code
when you don’t pay your
technical debt

How to deal with Technical Debt



What SonarQube is / does

- Free & open source “Code Quality Platform”
- Provides moment-in-time quality snapshots
- Gives trends of lagging and leading indicators
- Tracks developers’ seven deadly sins (seven axes of quality)

How does it work?

- Analyzes source code and byte code
- Computes hundreds of metrics
- Associates metrics with analysis snapshots
- Shows the results in dashboards and widgets accessible by any browser

SonarQube for everything

- Initially developed only for Java projects
- Today supports over twenty languages

Commercial : ABAP, C, C++, Cobol, Natural, PL/SQL, Visual Basic

Open Source : C++, C#, Flex, Groovy, Android, Javascript, PHP, Python, XML, Web(xhtml, jsp , jsf,)

… and for everyone

For developers. Is my code "good"? How can I improve it?

For testers / QA staff. Which parts of the system lack unit testing?

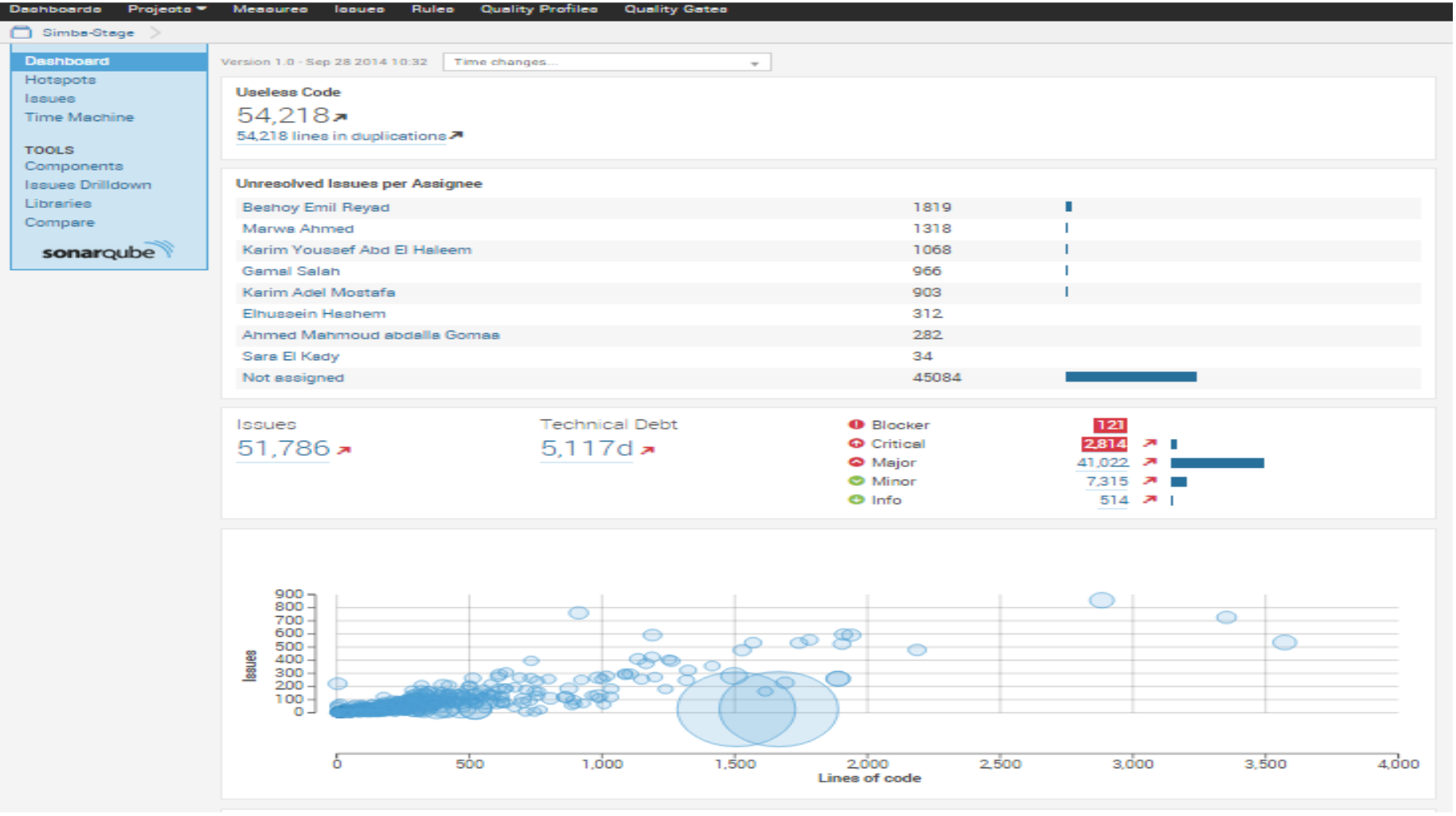
For architects. Is the initial design "broken"? How about complexity?

For managers. Give me the numbers!! Are we going up or down?

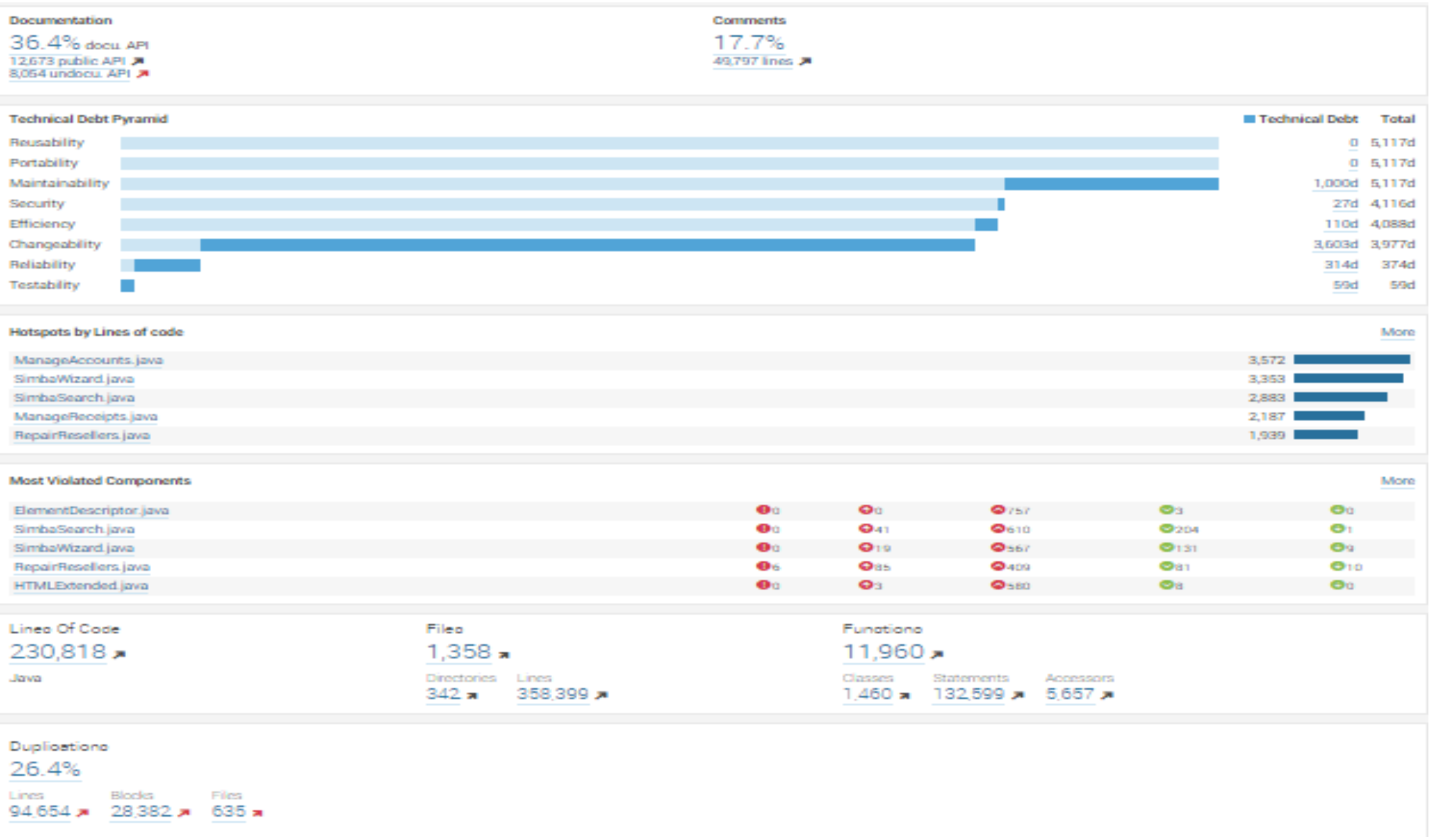
Managing code quality

- Dashboards
- Historical data
- Differential views
- Compare service
- Code reviews
- Action plans

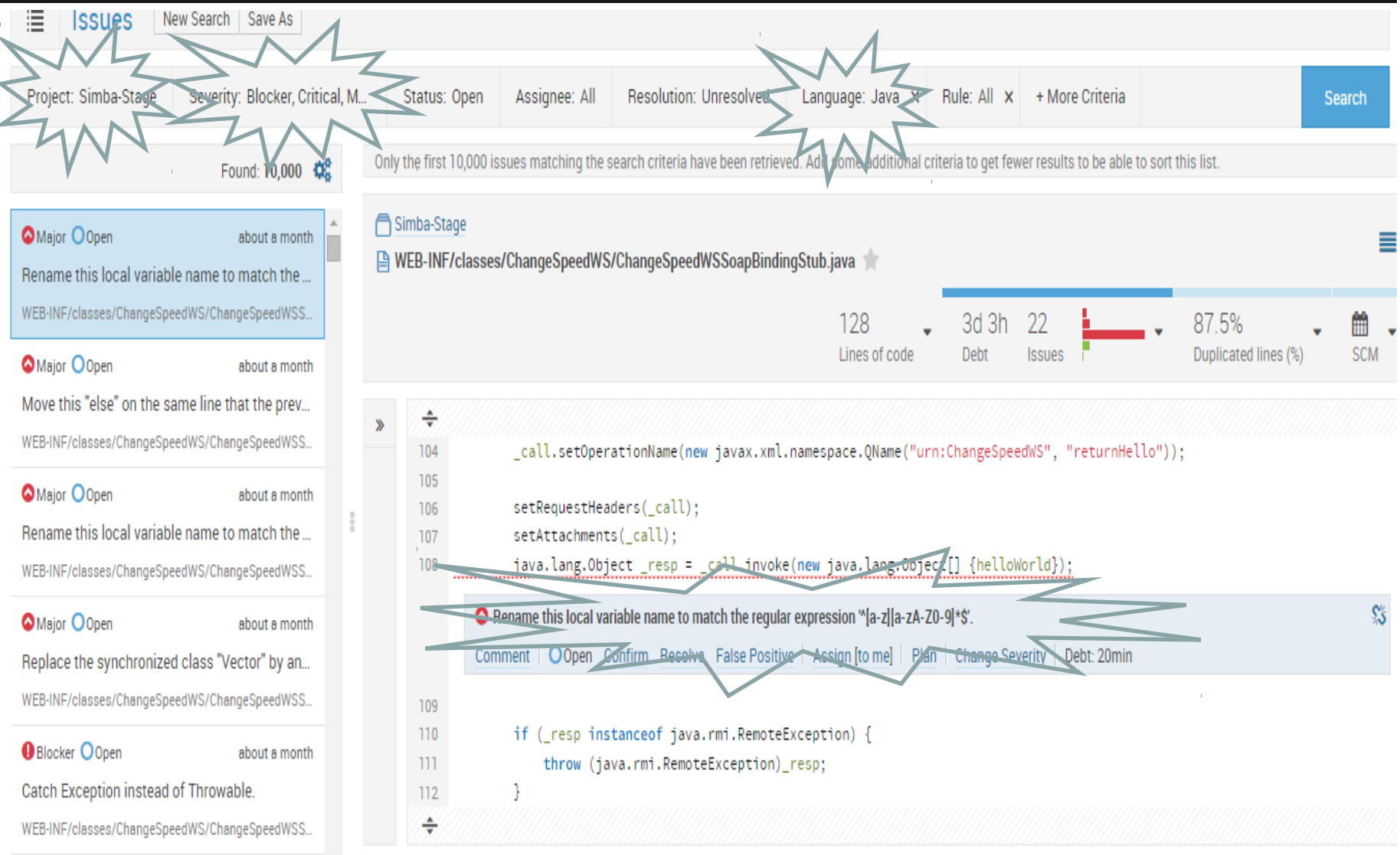
Managing code quality



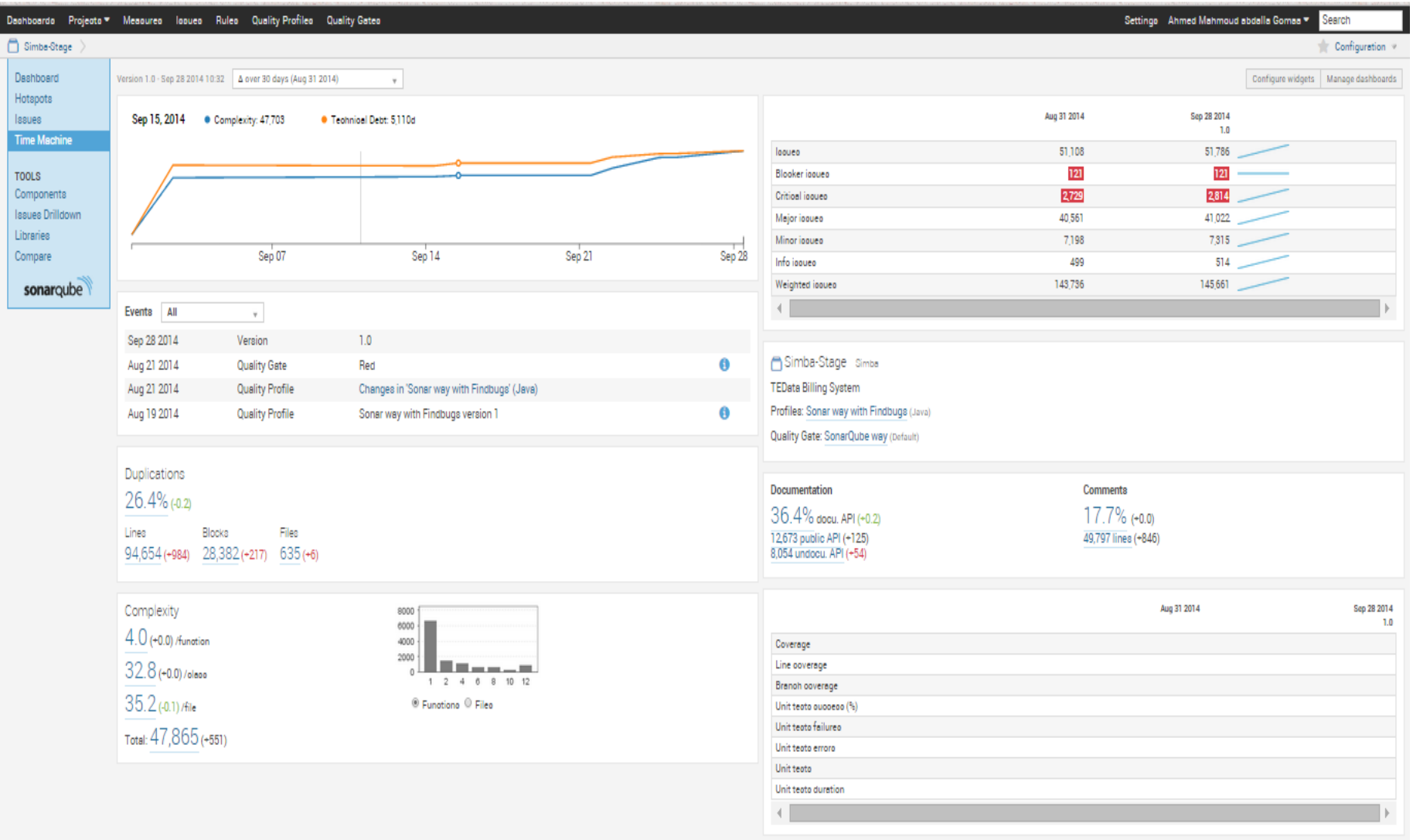
Sonar – Basic statistics



Sonar – Project Drill Down



Sonar – Time Machine



The big picture

- Track and reduce Technical Debt on an ongoing basis. (Clean up kitchen every day)
- Engage all developers from project day #1 (Not only mums wash the dishes)
- Get alerted when Technical debt is beyond a threshold (when someone is leaving the kitchen in a mess)

I have a dream…

...that one day code quality management will be as much as important and essential is today source code management

Questions

Refrences

Tracking & Improving S/W Quality with SonarQube by Patroklos Papapetrou)