

# Continuous Product-Focused Project Monitoring with Trend Patterns and GQM

Hidenori Nakai, Kiyoshi Honda,  
Hironori Washizaki, Yoshiaki Fukazawa (Waseda Univ.),

Ken Asoh, Kaz Takahashi, Kentarou Ogawa, Maki Mori,  
Takashi Hino, Yosuke Hayakawa, Yasuyuki Tanaka,  
Shinichi Yamada, Daisuke Miyazaki (Yahoo Japan)

Presenter: Hirohisa Aman (Ehime University)

# Background: Importance of understanding by stakeholders

- It is important for **project stakeholders** to understand the **state of projects** and the **quality of products**



Is the testing sufficient?

Were most of defects detected and fixed ?

- A proper understanding helps a **cost saving** and effective **process/quality improvements**

# Background: Software Metrics

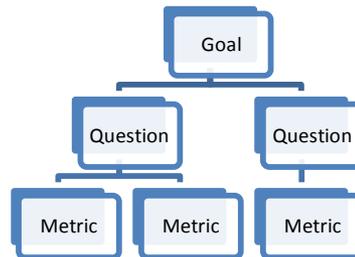
- **Software metrics** are often used for observing the state of projects, quality of products, and their tendencies **in quantitative manners**



- The importance of quantitative management with using metrics has been widely known (e.g. CMM/CMMI)

# Background: GQM

- It's **not easy for project stakeholders** to select appropriate metrics
- A Helpful method is **Goal-Question-Metric (GQM)**:



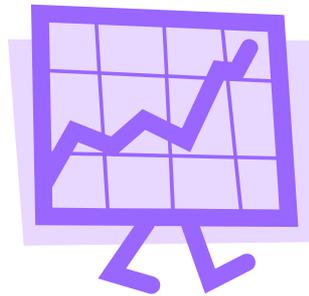
(Ex.)

- Goal: To evaluate the inspection process
- Question: How effective is the inspection process?
- Metric: Average faults detected per KLOC

# Proposals in this paper

The authors propose

- an approach to recognize **tendencies of projects and products**, and
- **Trend Patterns** to recognize tendency of project states and product quality

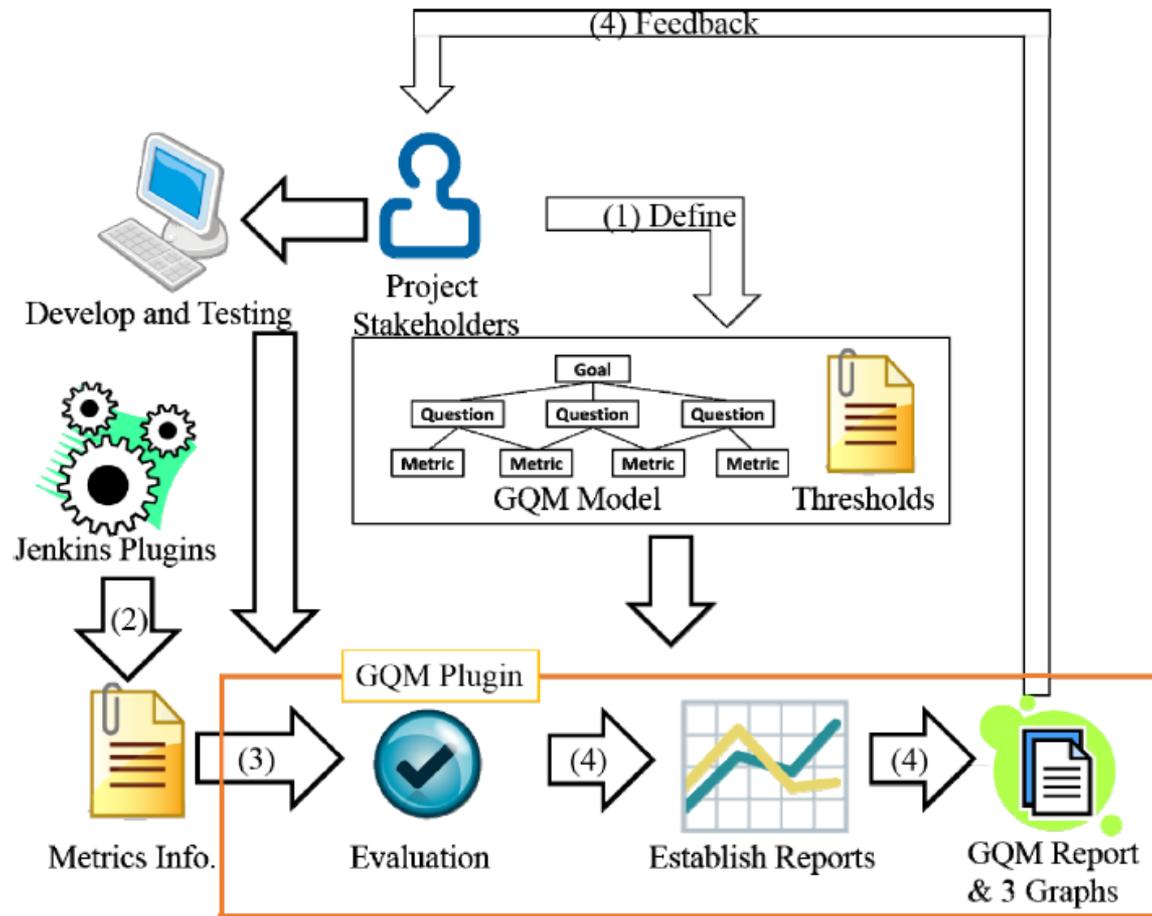


# Tool, Experiment and Case Study in this paper

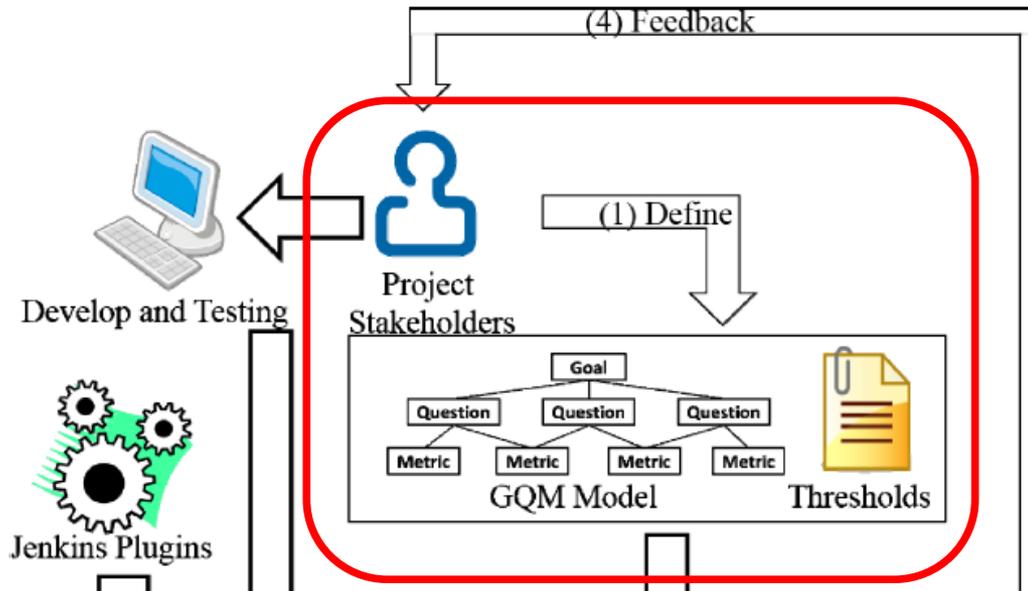
The authors

- developed a tool named **GQM Plugin** as a **Jenkins plugin**, to **visualize** evaluation results and tendencies of projects and products,
- conducted an experiment with an **OSS product**, and
- performed a case study in two **industrial development projects**

# Approach in this paper



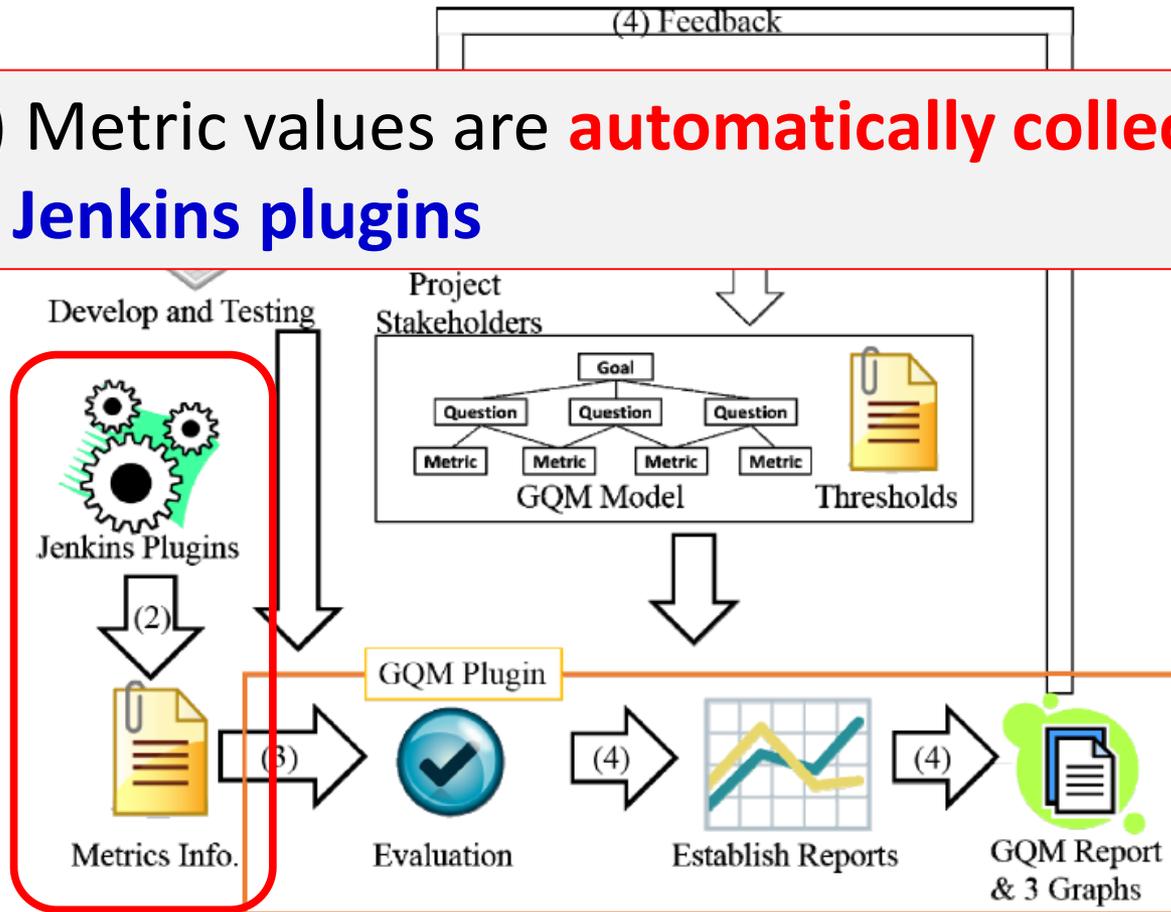
# Approach in this paper



(1) Project stakeholders prepare **metrics** that are appropriate to their project goals **through GQM model**, and define their **thresholds** of metrics to decide whether **they should pay attention** or not

# Approach in this paper

(2) Metric values are **automatically collected** by **Jenkins plugins**



# Data(Metrics) collected in the scheme

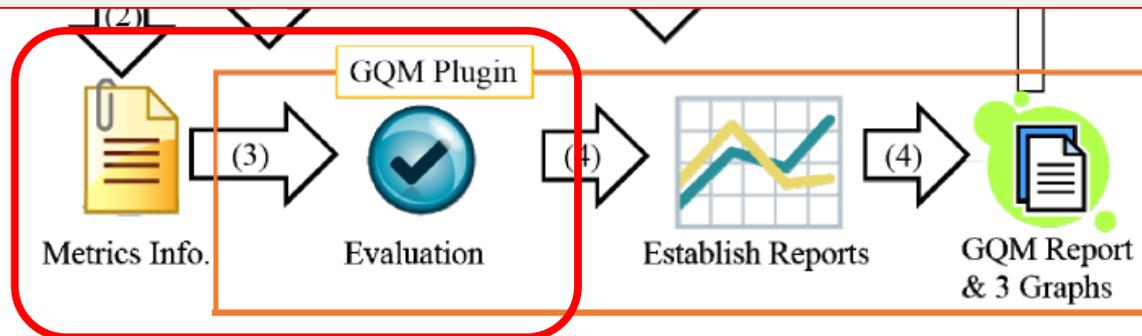
- Cobertura Plugin, Clover Plugin, Checkstyle Plugin and Reliability Plugin are used to collect
  - Test coverage
  - LOC
  - The number of coding standard violations
  - The number of violations of JavaDoc
  - open/close issues managed at GitHub
  - Predicted issues and the number of potential defects

# Approach in this paper

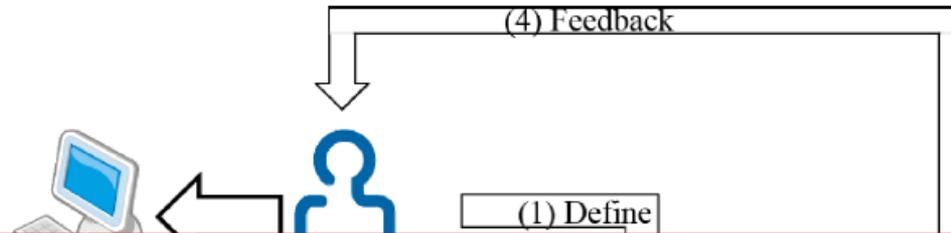
(3) Project status and product quality are **evaluated** based on the **GQM model**, the **metric values** and **their thresholds** (in GQM plugin).

Evaluations are

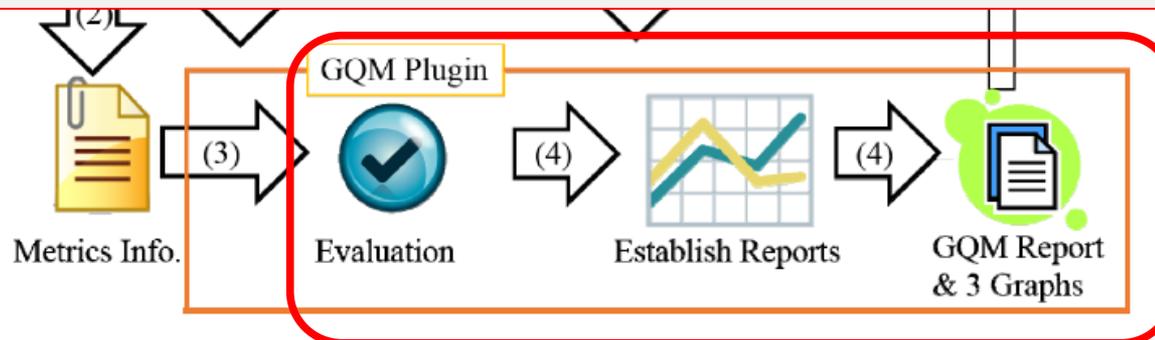
- **Error**: the metric value is unacceptable; to be improved as soon as possible
- **Warning**: almost OK, but not all
- **Normal**: OK



# Approach in this paper



(4) The **GQM report** and **trend graphs** are outputted by GQM plugin; the trend graphs show **changes** in **metric values** and in **evaluations of metrics and questions**



# Trend graphs

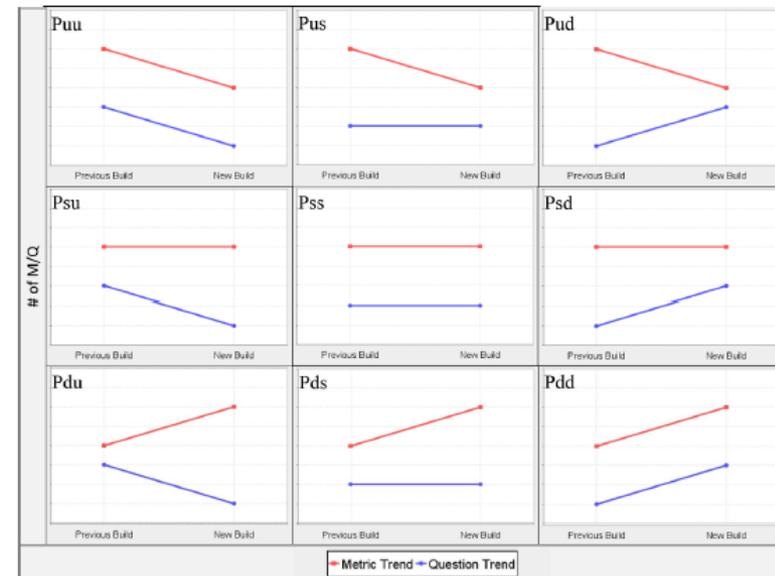
- **Metrics Trend Graph (MTG)**, and
- **Question Trend Graph (QTG)**:  
Numbers of metrics and questions evaluated as “Error”, “Warning”, or “Normal”
- **Metrics Value Trend Graph (MVTG)**:  
Changes in metric values during the development (by builds of the system)

If some metrics are increasing/decreasing and they are evaluated “Error” or “Warning,” project stakeholders can see that they should be improved.

# Trend Patterns

		Question evaluation		
		Up	Stable	Down
Metric evaluation	Up	Puu	Pus	Pud
	Stable	Psp	Pss	Psd
	Down	Pdu	Pds	Pdd

- Up = getting better
- Stable = no changed
- Down = getting worse



# Case Study

- Evaluation experiments are conducted for an open source project and two closed commercial projects

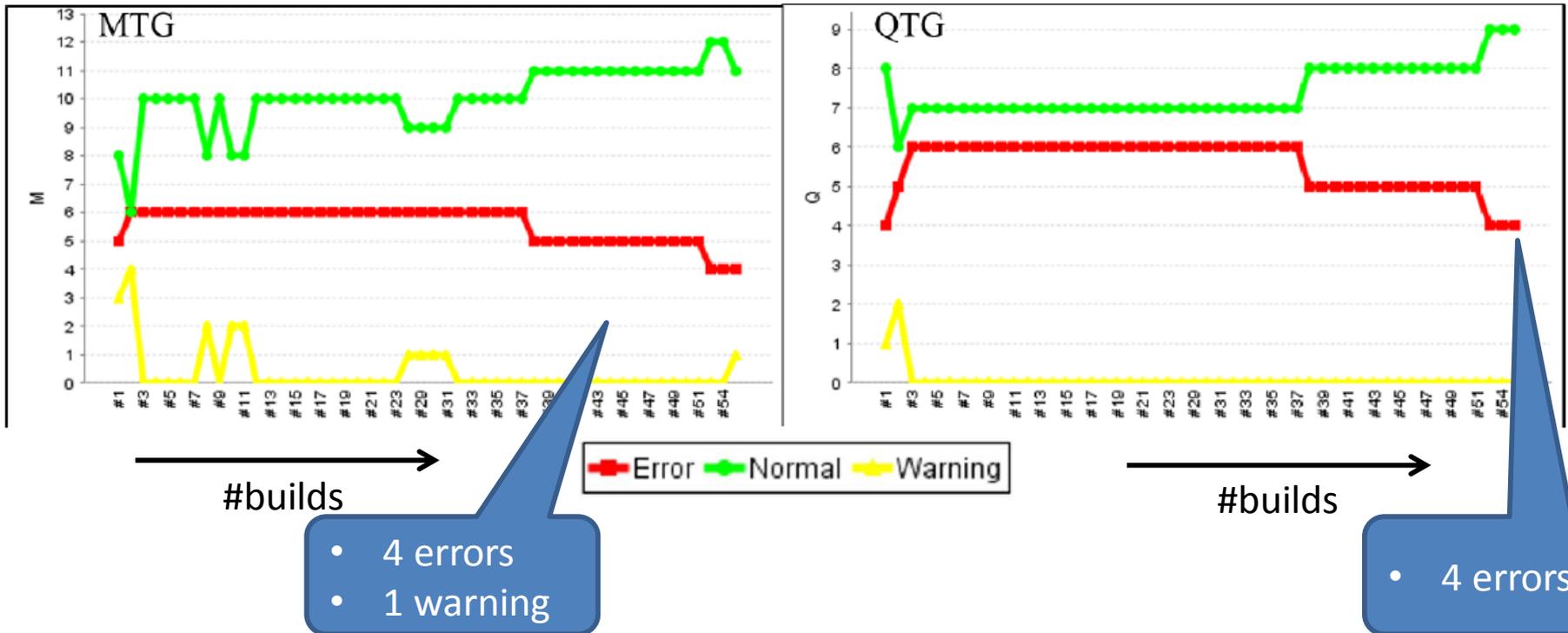
Name	Maven-android-plugin 6	Project A	Project B
Domain	Maven plugin for android application development	Platform (Web software development)	Library (Web software development)
Language	Java	Java, PHP	C++, Scala

# GQM model

- **Three (3) goals**
  - To ensure functionality,
  - To ensure maintainability, and
  - To decrease remedy time for bugs in source code
- **Thirteen (13) questions**
  - Are there enough tests on important modules?
  - ....
- **Twenty-one (21) metrics**
  - Test coverage
  - ....

# Result: Maven-android-plugin 6

- MTG (Metrics Trend Graph)
- QTG (Question Trend Graph)



# Result: Maven-android-plugin 6

- According to MVTG of this project, the authors identified that **test coverage is not enough** in each build
- Error metrics are “**Stay time**” and “**Test coverage**”
- Warning metric is “**The number of potential defects**”
- Error questions are “**Are there enough tests on important modules?**”, “**Are there enough tests on modules which have middle/low importance?**” and “**Are the defects corrected quickly?**”

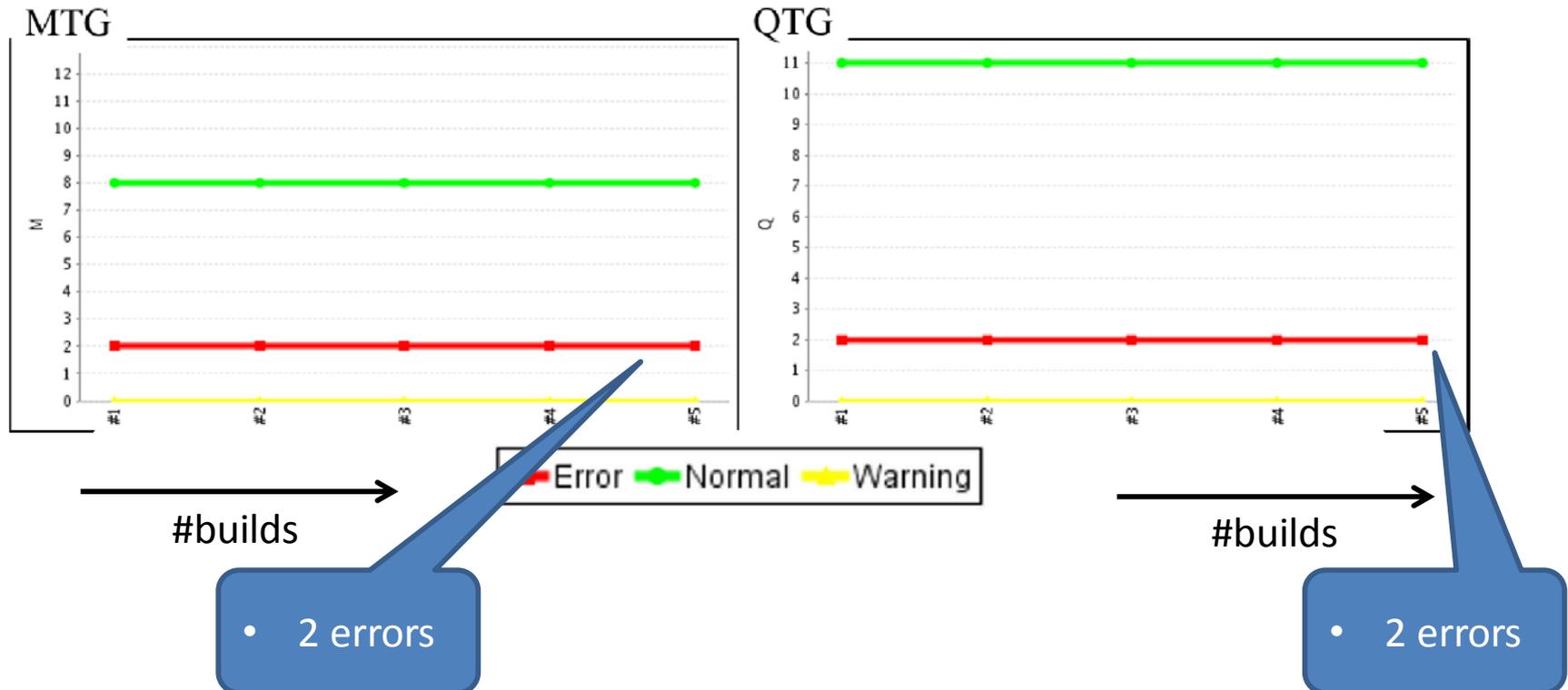
time from detection to correction of defects

# Discussion: Maven-android-plugin 6

- The number of **error** metrics and questions are **decreasing**: its pattern is **Puu**
- While this project's state and product quality are improved, some errors remains
  - The **test is not enough** and **time from detection to correction of defects is too long**
  - Project stakeholders should improve test coverage

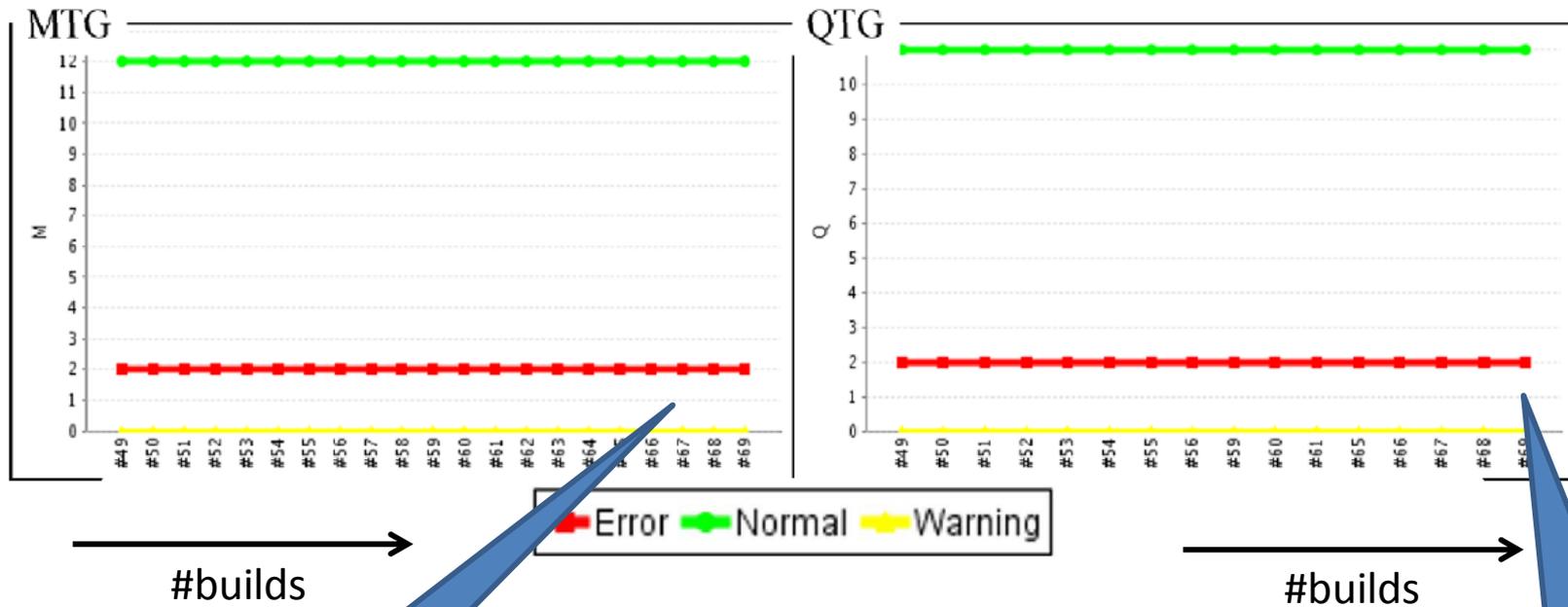
# Result: Project A

- MTG (Metrics Trend Graph)
- QTG (Question Trend Graph)



# Result: Project B

- MTG (Metrics Trend Graph)
- QTG (Question Trend Graph)



• 2 errors

• 2 errors

# Results: Project A & B

For both of projects,

- Error metric is “**Stay time**”
- Error question is “**Are the defects corrected quickly?**”

# Discussion: Project A & B

- The number of **error** metrics and questions **did not change**: its pattern is **Pss**
- However, all modules of these projects have same error: **the time from detection to correction of defects is too long**
- On the other hand, **no other problems** are found

# Conclusion

- An application of **GQM model** and a **time-series analysis of metrics** with **trend patterns** are proposed
- **GQM plugin** is developed and utilized
- Evaluation experiments are conducted for an open source project and two closed commercial projects

# Conclusion

- For each project, the proposed method **helps to identify problems** that **stakeholders should notice**:
  - e.g., insufficient tests, long time from detection to correction of defects, and potential defects

# Future Work

- To integrate the approach and the GQM Plugin to software development project continuously
- To adopt them to other domain projects
- To extend the GQM Plugin to collect more kinds of metrics information (e.g., cohesion, coupling, and code clone)