Asian Software Quality Network Conference 2017 凱旋講演会

Metrics Extraction and Evaluation in Enterprise Software Development using OSS

24th November 2017

Sumie Moita[†1] Shinji Kikuchi[†1] Takuto Mizutani[†1]
Kazuki Munakata[†1] Masako Ito[†2] Masao Tsuchiya[†3]
Susumu Rikitake[†3] Tatsuki Kobayashi[†3] Makoto Nonaka[†4]

[†1] FUJITSU LABORATORIES LTD.
[†2] FUJITSU LIMITED
[†3] Fujitsu Computer Technologies
[†4] Toyo University

Table of contents

1. Background: Problems in use of OSS in enterprise software development
2. Metrix extraction: Questionnaire survey
3. Validity of metrics
4. Conclusion

1. Background: Problems in use of OSS in enterprise software development

OSS in enterprise: Background and problems

Background

1. **Software development in large scale/short term**
   - Expected to reduce development workload by utilizing OSS
2. **Sometimes enterprises develop/enhance functions in OSS**
   - OSS communities might not develop functions required by certain customers

Problems

1. **Source is open, but virtually black-box**
   - When a problem happens, difficult to determine if its cause is in OSS or not.
2. **Heavy workload to keep up with OSS version-up**
   - Using latest software or extend modules

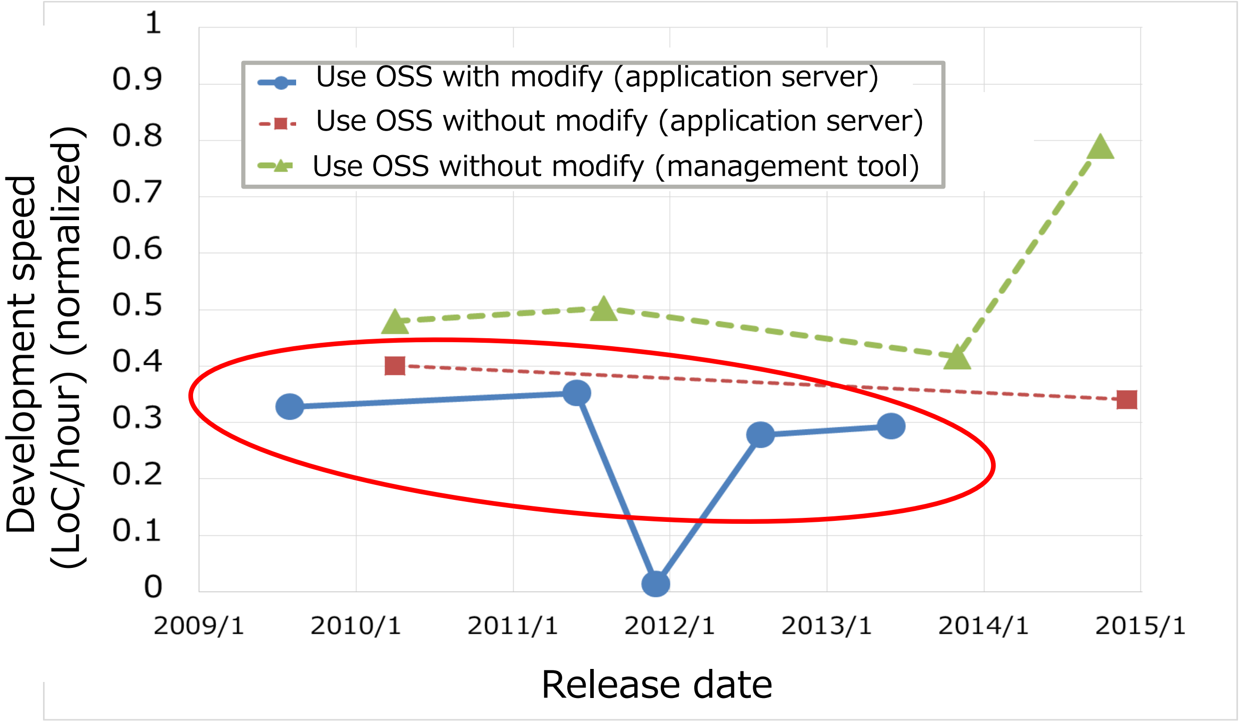
OSS usages in development in enterprises

	No proprietary modification	With proprietary modification
Overview	- Use OSS's functions developed by communities. - OSSs have enough functions required for a system.	- Enterprises modify OSSs and use them in their products / services. - OSSs do not have enough functions and modifications are allowed by their license.
Function development	- By OSS communities	- By OSS communities and enterprises
Quality	- Depends on OSS communities	- Enterprises can control software quality ☺
Productivity	- Less development ☺	- Need to understand OSS and develop functions
Maintainability	- Less maintenance effort ☺	- Need to keep up with enhancement in OSS
Problem	- Enterprises cannot control the quality of OSSs	- Difficult to read / understand / modify OSS codes developed by communities

Less flexibility, less cost ⟷ More flexibility, more cost

Case study (1/2)

Development speed and OSS usage



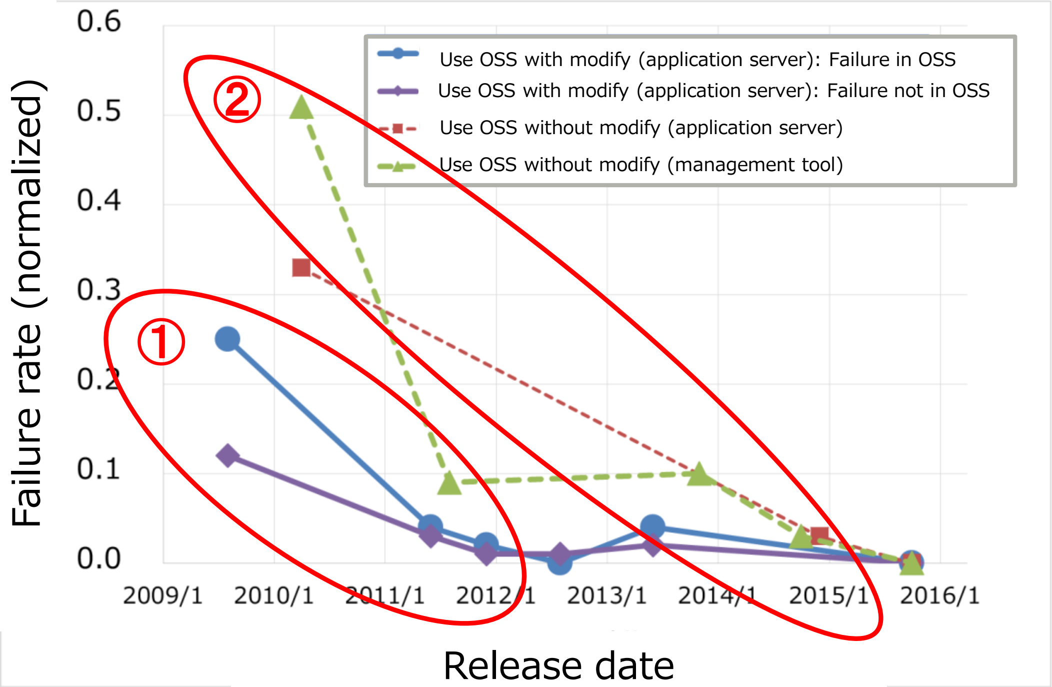
※ Release in 2012 was too small (exceptional value)

Speed

- ■Development speed is small with OSS modification.
- ■Number of releases does not improve the speed.

Case study(2/2)

OSS usage and quality



Quality

① With modification: Became stable in short time

② Without modification: Took long time to become stable

Summary

Tradeoff between quality and development speed in use of OSS with/without modification

Assumptions and Research Questions

FUJITSU

- 2 assumptions and tree research questions to understand the problems in OSS usage in enterprise software development

AS-1

Using OSS can degrade QCD in enterprise software development

AS-2

The main cause of QCD degradation is engineers not having enough skill

RQ-1

- Were there any QCD problem in projects using OSS?

RQ-2

- Did the QCD problems come from OSS?
- In which development phase did the problem occur?

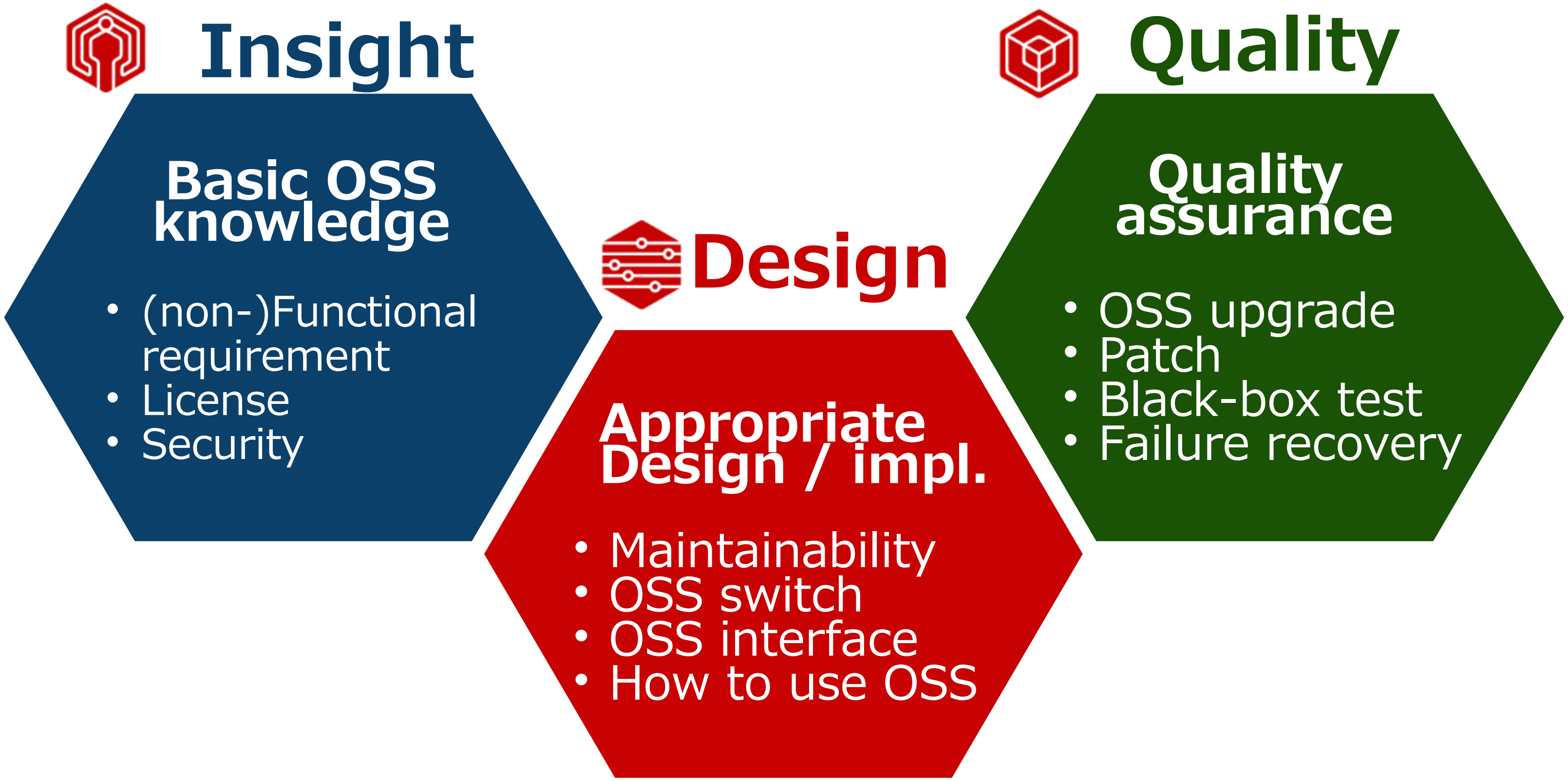
RQ-3

- What kind of problems do engineers think they have?

※ QCD: Quality, Cost, Delivery

Skills for software development using OSS

- 3 types of skills required in software development process utilizing OSS

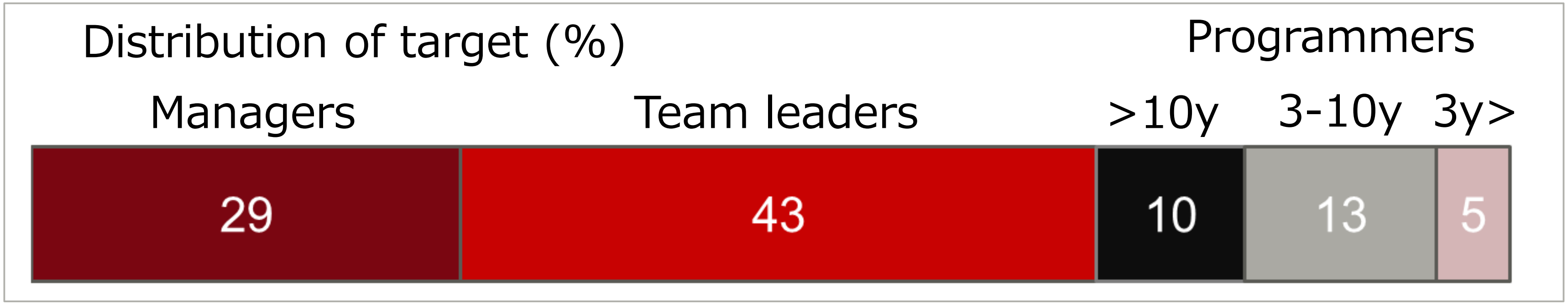


2. Metrics extraction through questionnaire survey

Questionnaire survey

Target

Development teams in an enterprise
130 engineers from 37 business departments



Types of used OSS（Top 3）	
Database (PostgreSQL, MySQL, etc.)	65
Web server (Apache, etc.)	63
Operating system (Linux, etc.)	59

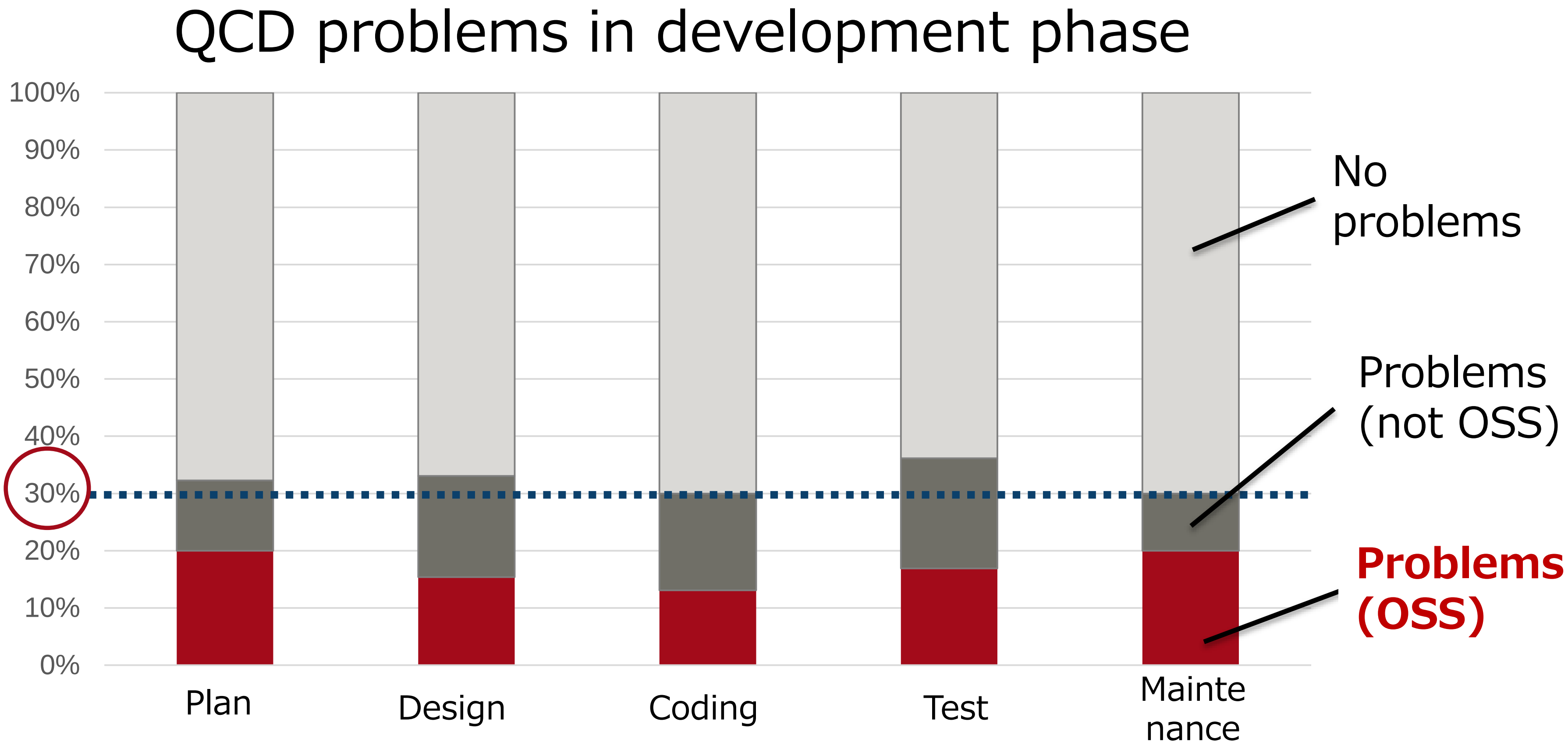
Questions

1. Were there any QCD problem in projects using OSS?
2. Were the QCD problems OSS-oriented?
3. What kind of problems do engineers think they have? (Following page)

Questionnaire of problems in using OSS

Insight	**q1**	Shortage of information of OSS’s functional requirement
	q2	Shortage of information of OSS’s nonfunctional requirement
	q3	Shortage of information of OSS’s intellectual properties
	q4	Shortage of information of OSS’s version compatibility
	q5	Information sharing in OSS usages in a company
Design	**q6**	Designing method to utilize OSS
	q7	Designing system considering maintenance and changes in OSS
	q8	Implementation by understanding OSS’s functions and interfaces
Quality	**q9**	Establishing quality assurance process in using OSS
	q10	Workload of testing black-box OSS
	q11	Keeping up with version up and security patch
	q12	Quality assurance for OSS itself
	q13	Failure analysis in system using OSS
	q14	Quality assurance in using several OSSs
Others	**q15**	Sharing experience of OSS usage
	q16	Rapid development by using OSS
	q17	Unexpected problems in using new OSSs
	q18	Learning new OSS in short time
	q19	Conflict between business and OSS contribution
	q20	Range of contribution to community

RQ-1: QCD problem in projects using OSS?



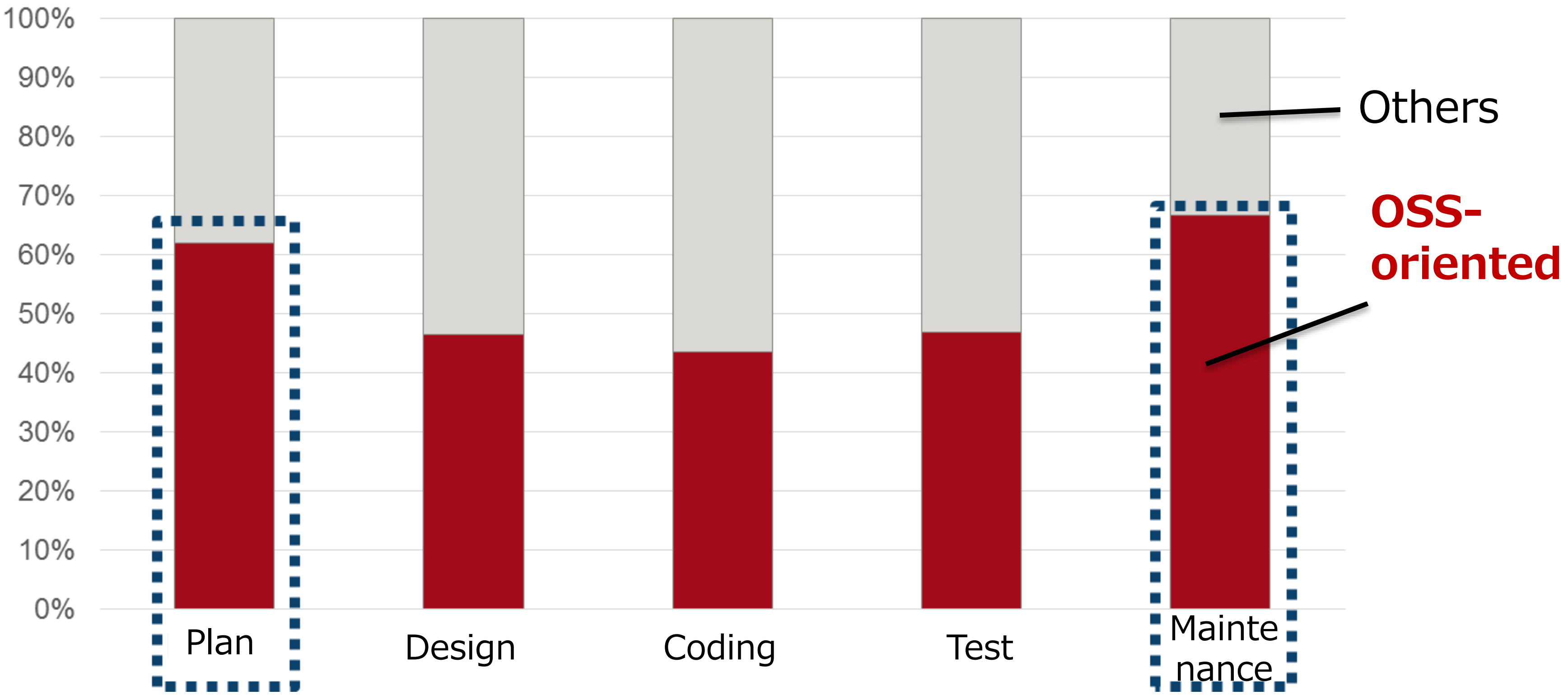
RQ-1

RQ-1 result

■30% QCD problem throughout development process

RQ-2: QCD problems in development phase

Cause of QCD problems in development phase



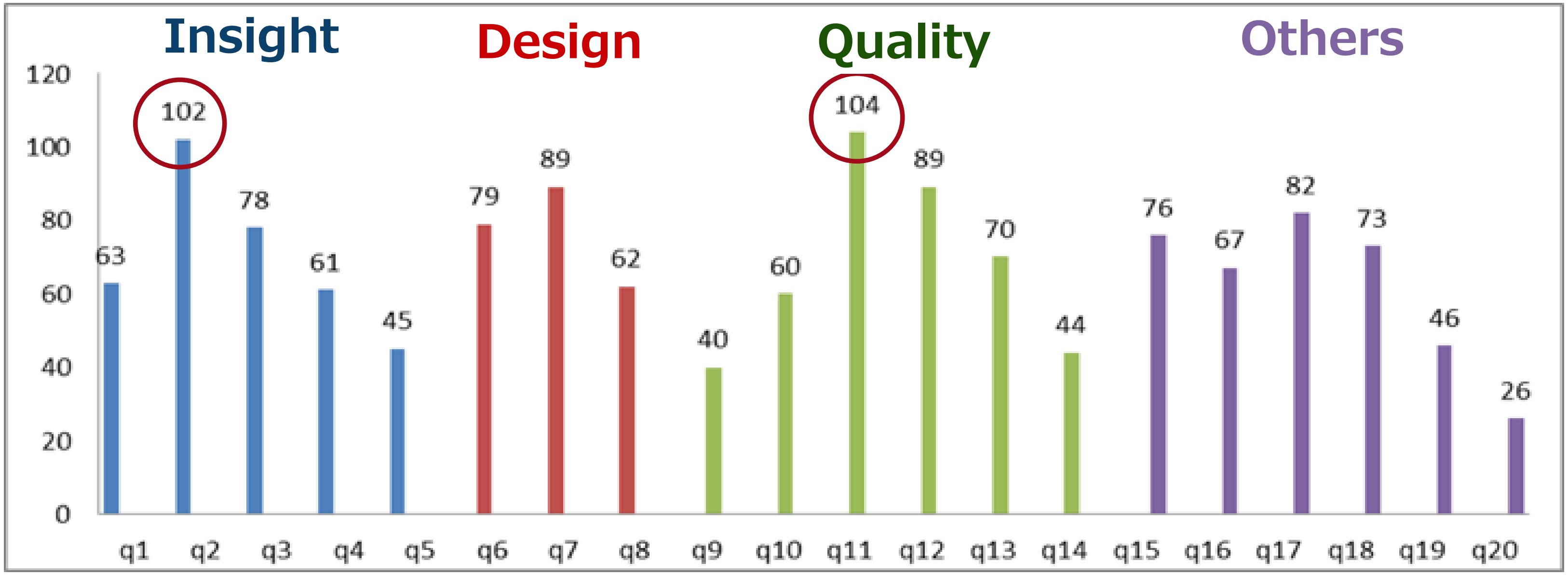
RQ-2

RQ-2 result

■OSS-oriented QCD problems in planning and maintenance phase are larger than other phases.

RQ-3: Types of OSS problems (1/2)

Problems in using OSS



Most frequent Answers (> 100)

Insight — **q2:** Shortage of information of OSS's nonfunctional requirement

Quality — **q11:** Keeping up with version up and security patch

RQ-3: Types of OSS problems (2/2)

FUJITSU

χ^2 test for OSS usage problems and stages

		Questions				
		q8	q9	q10	q12	q18
Phase	Plan	**0.006****	0.999	**0.003****	0.308	**0.050***
	Design	**0.004****	0.500	0.647	**0.044***	**0.023***
	Test	0.496	**0.007****	0.913	0.327	0.162
	Maintenance	**0.047***	1.000	1.000	0.026*	0.720

**(p<0.01); *(p<0.05)

RQ-3

RQ-3 result

- q8(understanding OSS's IF and functions) gives impacts widely (plan, design, and maintenance)
- Plan phase has relation with q8, q10(difficulties in testing for black-box OSS) and q18(learning new OSS in short time)

Current metrics in enterprise development

Category	Metrics	Detail	Unit
Insight	Information from OSS communities	Amount of source codes	LOC
		Amount of documents	-
		Version information	-
		Code metrics	-
		License	-
	Quality	Bugs recorded in BTS	(int) Bugs
		Time to recover	Days
		Patch release frequency	Days
	Vulnerability	Number of vulnerabilites	(int) Vulnerabilities
		Severity	Critical/Major/Minor
Design	Development size	Steps developed internally	LOC
		Size of OSS being used	LOC
	Cost (person-hour)	Design, impl., test, maintenance	Person-hour
		Time for reviews	Hour
Quality	Quality of parent soft	Number of remained failures	(int) Failures
	Quality of design	Failures found in reviews	(int) Failurs
		Severity of failures	Critical/Major/Minor
	Quality of test	Number of identified bugs	(int) Bugs
		Severity of failures	Critical/Major/Minor
		Number of degradation	(int) Degradation
	Quality after release	Failures after release	(int) Failures

Extracted metrics（OSS utilization）（1/2）

FUJITSU

Category	Metrics	Detail	Unit
Insight	Community activity	Comments from users	Comments / month
		Number of users	(int) Person
		Developer activities	Developer comments
		Number of developers	(int) Person
	Public (social) data	Bluckduck score	Points
		Support	-
		Project continuity	Years
		Web search hit	(int) hit
		Slideshare	(int) slides
	Record in an enterprise	Information of OSS, products, etc.	-
	OSS engineers In a team	Number of OSS specialists	(int) Person
		Number of committers	(int) Person
		Number of commits	(int) Commits

Extracted metrics（OSS utilization） (2/2)

Category	Metrics	Detail	Unit
Design	Development size	Size of OSS used in product	%
	Functions	Number of provided functions	Functions / Person-hour
	Development speed	Release frequency	Times / year
	Planning	Diversion from plan	Hour
Quality	Test code size	Test code steps	LOC
	Test automation rate	Test execution automation ratio	%
		Test result evaluation automation rate	%
	Quality of base soft	Backlog	-
		Remained bugs in OSS	(int) bugs
	Code metrics	Coverage	%
		Number of nests	(int) nest
		Complexity	Complexity degree
	Process maturity	Maturity of non-functional requirement	(int) requirements
		CI results or CI success rate	-

3. Validation of metrics

Evaluation of OSS insight

- Metrics for OSS insight
  - Number of engineers concentrating on OSS development,
  - OSS community participation rates
- Relation between metrics and QCD problems

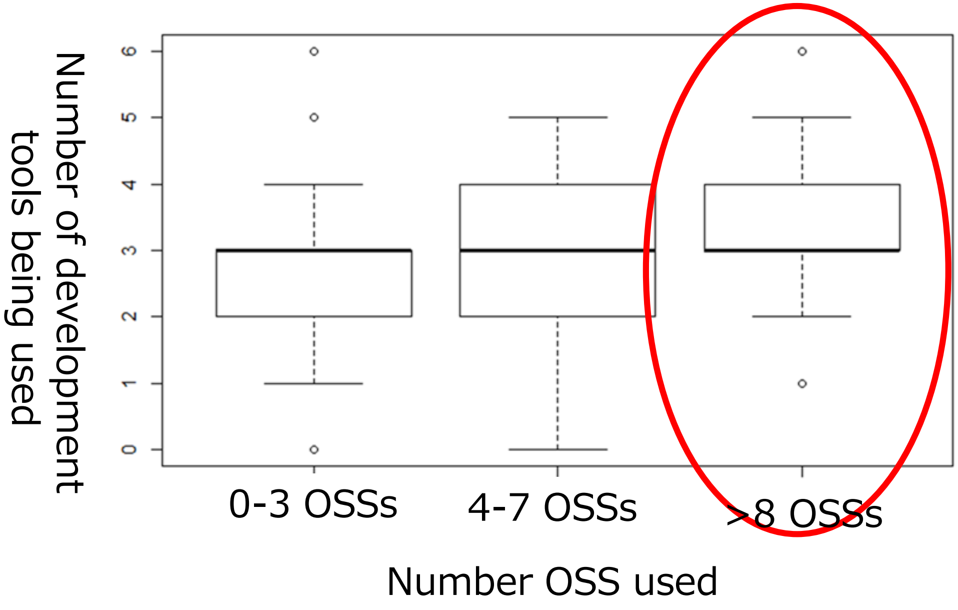
Metrics (in a team)	OSS-oriented QCD problems?		Overall Average
	Yes	No	
Number of OSS developers (%)	8.9	3.7	6.0
Number of OSS community members (%)	1.4	0.9	1.1

OSS insight

- Teams having OSS developers/community members can notice OSS-oriented QCD problems.

Evaluation of design skill with OSS

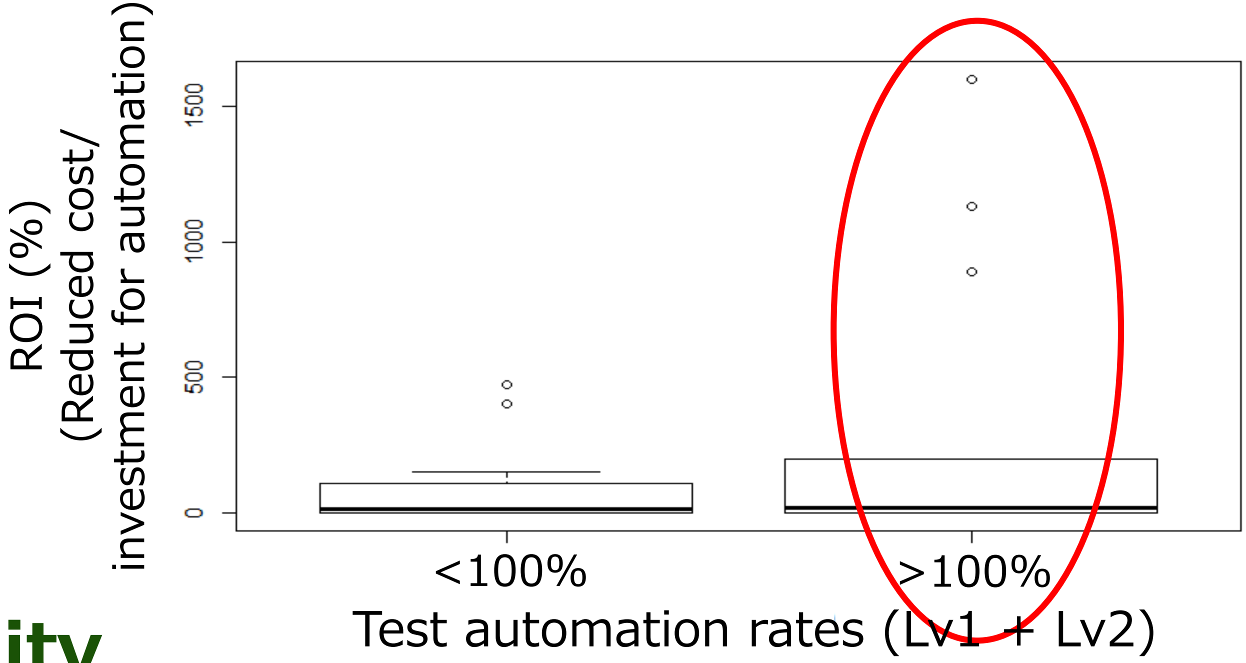
- Metrics for design skill with OSS
  - Number of OSS in product
- Number of tools and OSSs used in development



- Teams trying to use many (new) OSSs utilize several development tools.

Evaluation of quality assurance skill

- Metrics for quality assurance skill
  - Test automation rates (LV1: Test only, LV2: test result check)
- ROI(Return of Investment) of test automation



Quality

- High test automation rates achieve high ROI.

4. Conclusion

Conclusion

Identify the problems in using OSS in enterprise

■Target : Development team in enterprise

■Purpose : To clarify how we can improve cost and development speed

3 types of metrics for development in using OSSs

■**Insight** : Number of developers, community members, etc.

■**Design** : Number of development tools, OSSs in product etc.

■**Quality** : Test automation rate, etc.

Evaluation of metrics

■OSS community members can find OSS-oriented problems

■Capability of using many OSSs helps using various tools

■Penetration of test automation improves quality and ROI

FUJITSU

shaping tomorrow with you