

**KHMELNITSKY NATIONAL UNIVERSITY
SYSTEM PROGRAMMING DEPARTMENT**



ONTOLOGICAL APPROACH TO THE ASSESSMENT OF INFORMATION SUFFICIENCY FOR SOFTWARE QUALITY DETERMINATION

Tetiana Hovorushchenko

Actuality of the research

Software quality is the ability of the software to meet the stated and predicted needs when using under certain conditions [ISO 25010, ISO 25030].

One of the most important causes of poor quality of large software projects are the increasing the number of components (subsystems) and the interfaces between them, and uncontrolled complexity of software systems [CHAOS Report, The Standish Group International].

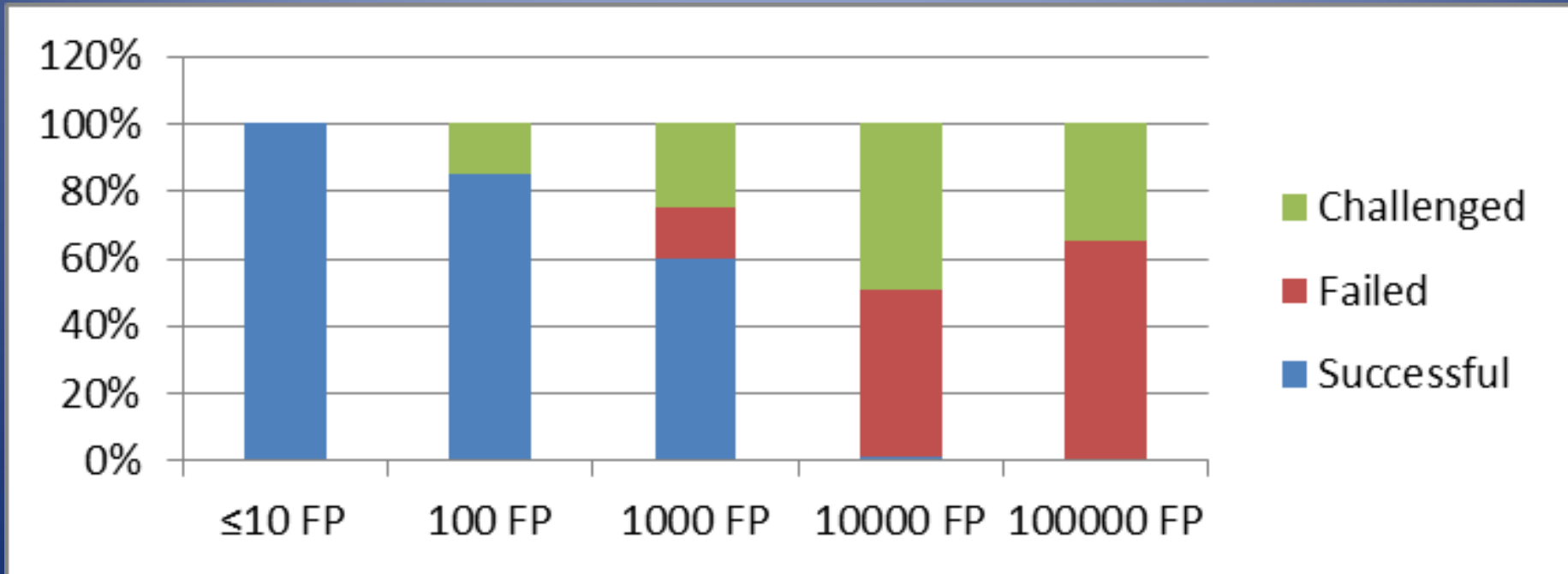
Research The Standish Group International shows that statistics of success of small, moderate, medium, large and grand software projects is significantly different:

CHAOS RESOLUTION BY PROJECT SIZE

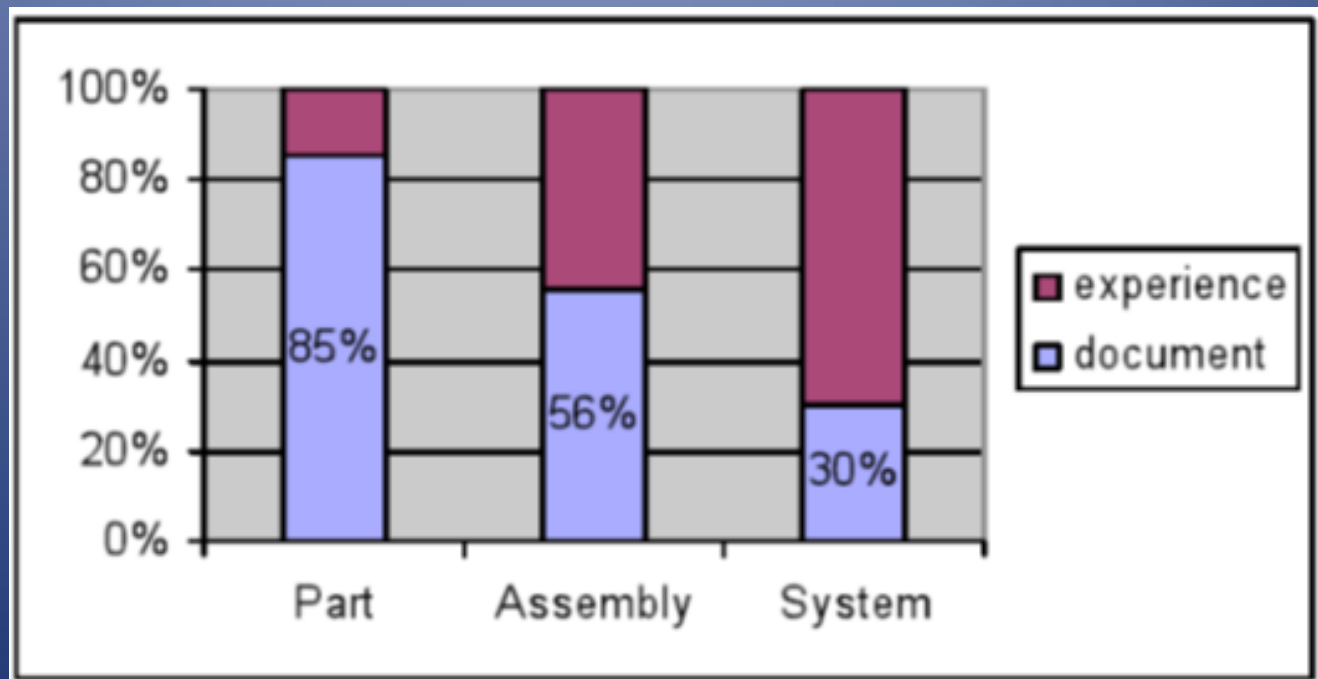
	SUCCESSFUL	CHALLENGED	FAILED
Grand	2%	7%	17%
Large	6%	17%	24%
Medium	9%	26%	31%
Moderate	21%	32%	17%
Small	62%	16%	11%
TOTAL	100%	100%	100%

The resolution of all software projects by size from FY2011-2015 within the new CHAOS database.

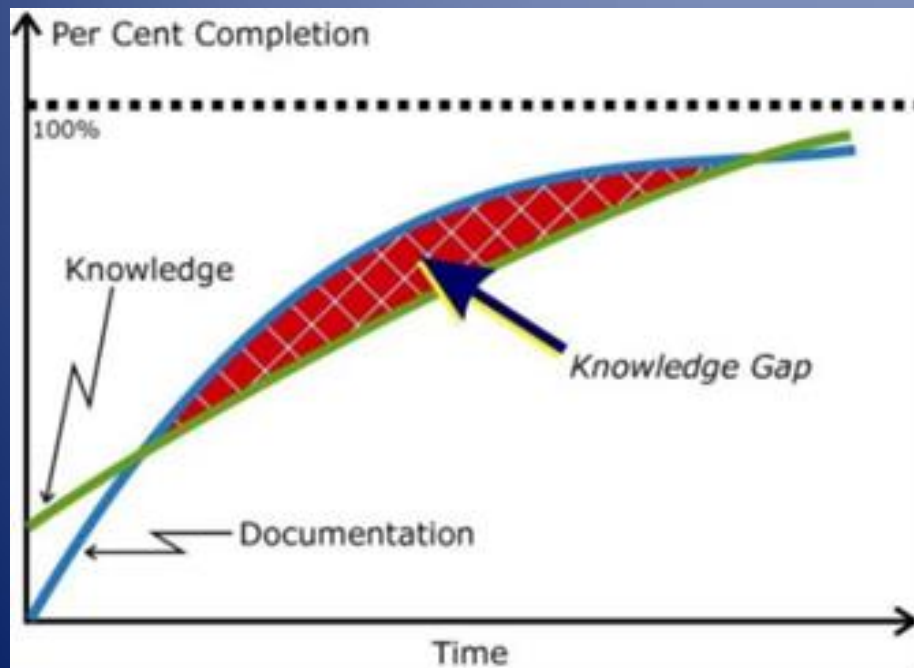
This conclusion is confirmed by the statistics of software projects **on the basis of the function points** as the main modern units of software size:



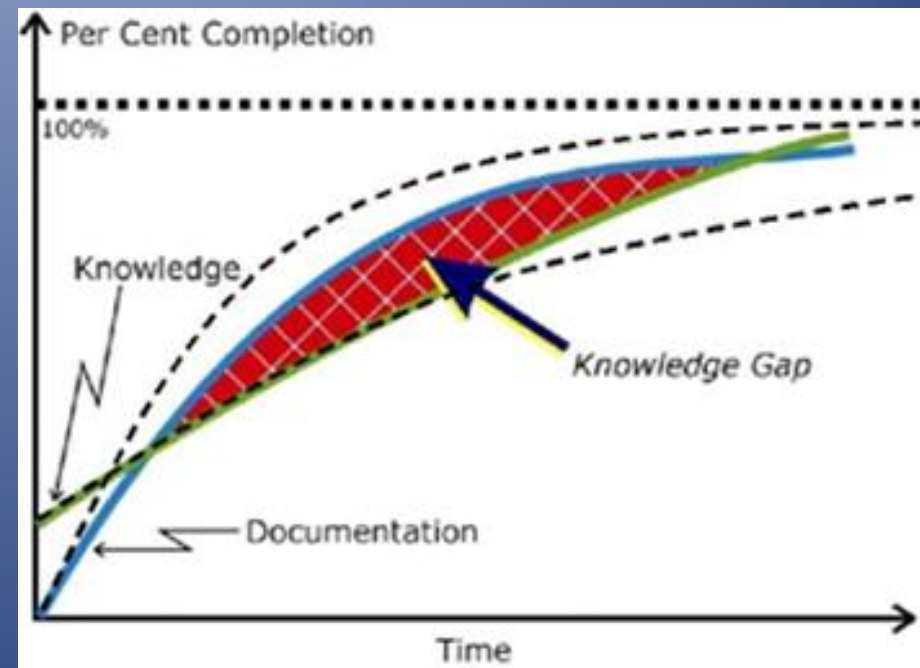
During the software project, we often can not estimate the share of the informational indeterminacy of the project. The cause of appearance of informational indeterminacy of the project is the **low level of knowledge documentation, especially at the system level :**



Now there is the situation, characterized by premature design decisions and their documentation, prior to understanding the design – area, referred to as the **“knowledge gap”** (the result of the low level of knowledge documentation and the root cause of many engineering failures):

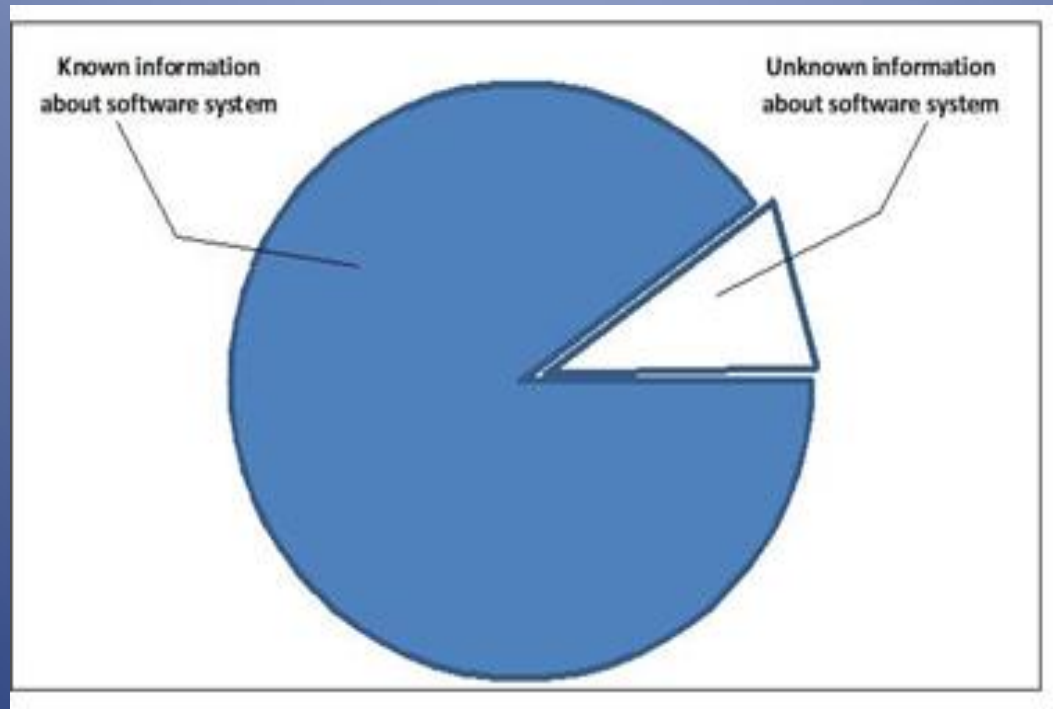


Patterson's vision



Our vision

All the available knowledge and information about the software system can be represented as the diagram, which has **the sector that reflects the volume of insufficient (unknown) information (knowledge gap):**

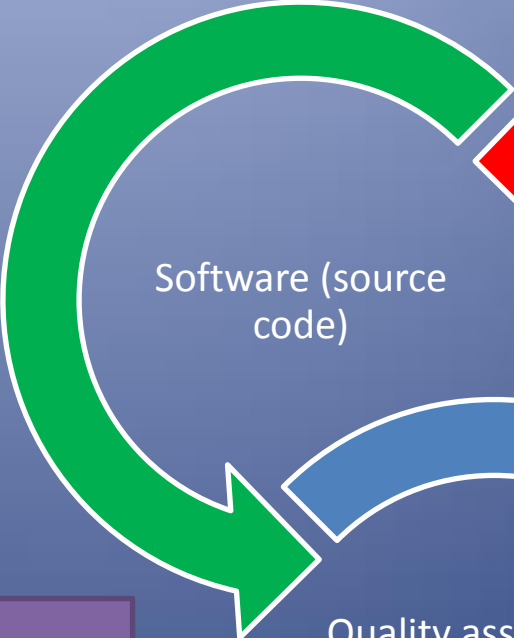
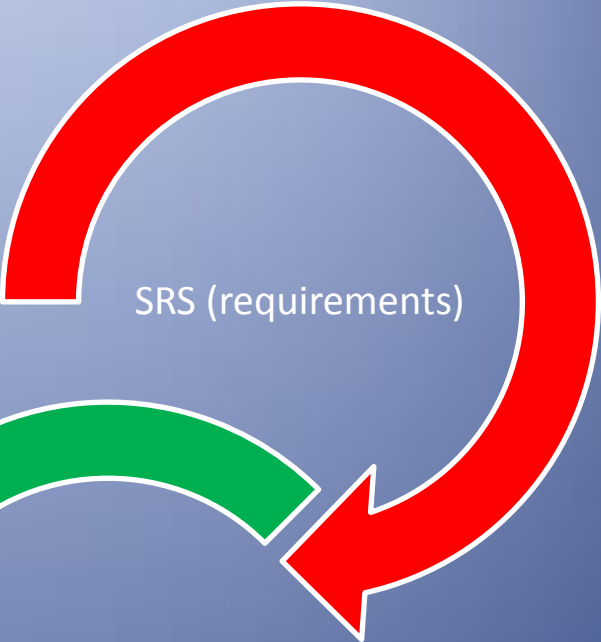


This sector consists of unconsidered subject domain information.

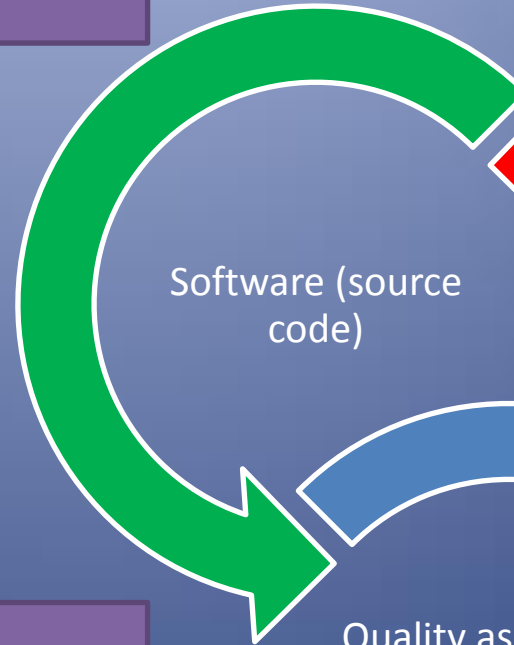
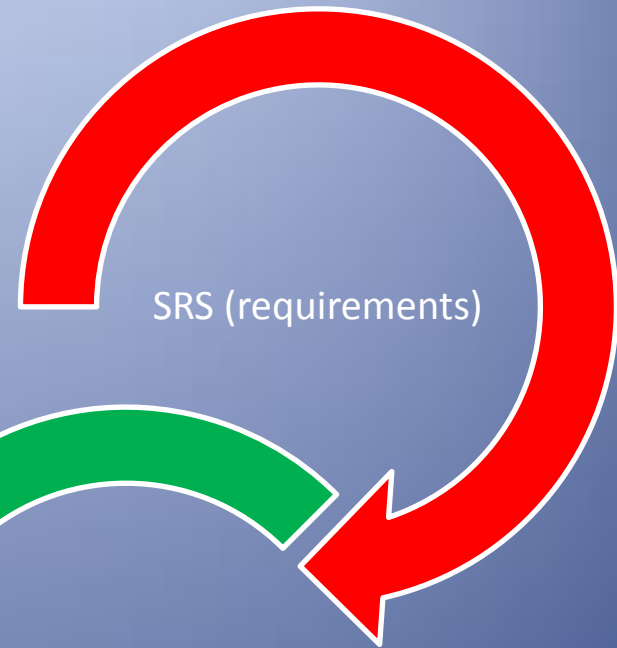
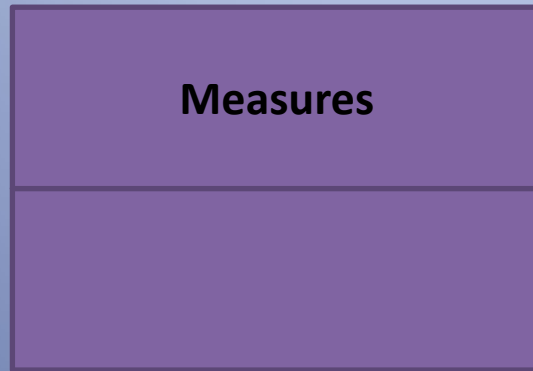
Then the **actual task** is the assessment of information sufficiency as to software (for example, the possibility of obtaining of trustworthy information on the measures for calculation of the values of the software quality characteristics and subcharacteristics), on the basis of which software quality (by ISO 25010) is determined. **Incompleteness and inaccuracy of such information lead to fall of veracity of software quality assessments.**

So the **purpose of this study** is the development of the approach to the assessment of information sufficiency for software quality determination.

Now



Proposition



Pair wise comparison of software quality characteristics for number of joint measures

	Functional Suitability	Reliability	Usability	Security	Performance Efficiency	Maintainability	Portability	Compatibility
Functional Suitability		2	2	2	2	3	3	3
Reliability	2		2	3	2	5	3	3
Usability	2	2		1	2	2	2	2
Security	2	3	1		2	3	2	2
Performance Efficiency	2	2	2	2		3	2	3
Maintainability	3	5	2	3	3		3	4
Portability	3	3	2	2	2	3		3
Compatibility	3	3	2	2	3	4	3	

Достовірність оцінок може бути підвищена за рахунок збільшення достатності обсягів інформації щодо атрибутів у специфікації вимог до ПЗ

Специфікація вимог до ПЗ (вимоги)

ПЗ (сирцевий код)

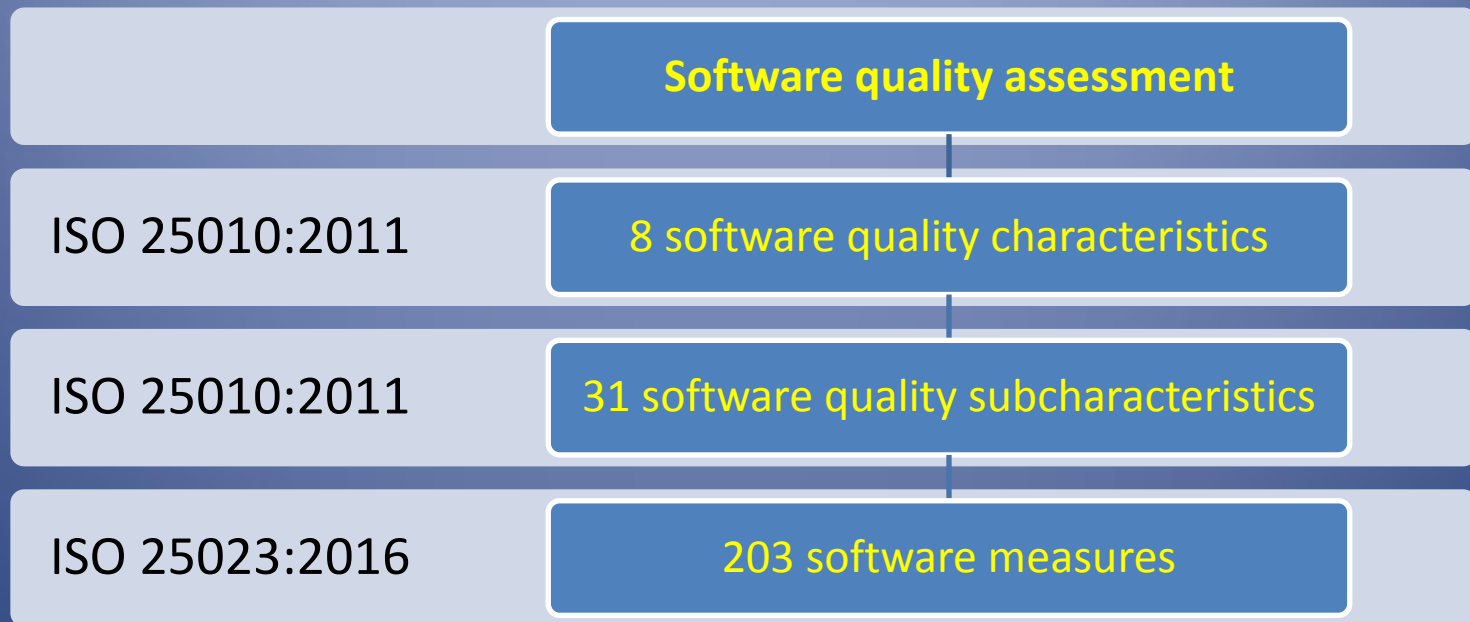
Оцінювання якості (атрибути, підхарактеристики, характеристики)

Оцінки недостатньої достовірності (через недостатність інформації щодо атрибутів)

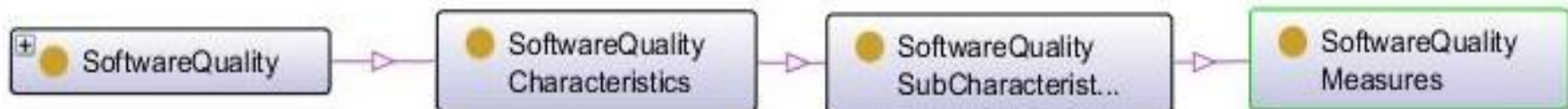
Спільні атрибути

Кореляція підхарактеристик та характеристик якості ПЗ

Ontological Approach to the Assessment of Information Sufficiency for Software Quality Determination



Idea of base ontology for subject domain “Software engineering” (part “Software quality”):



Ontological approach to the assessment of information sufficiency for software quality determination by ISO 25010:

1

- Development of the base ontology for subject domain “Software engineering” (part “Software quality”)

2

- Analysis of the SRS for the concrete software project for the presence of measures, that necessary for determining the quality characteristics and subcharacteristics of software quality

3

- Development of the ontology for concrete software project

4

- Comparison of the concrete ontology with the base ontology

5

- Identification of measures, which are absent in the ontology for determination of the quality of the concrete software

6

- Identification of quality characteristics and subcharacteristics, that cannot be calculated on the basis of the existing measures

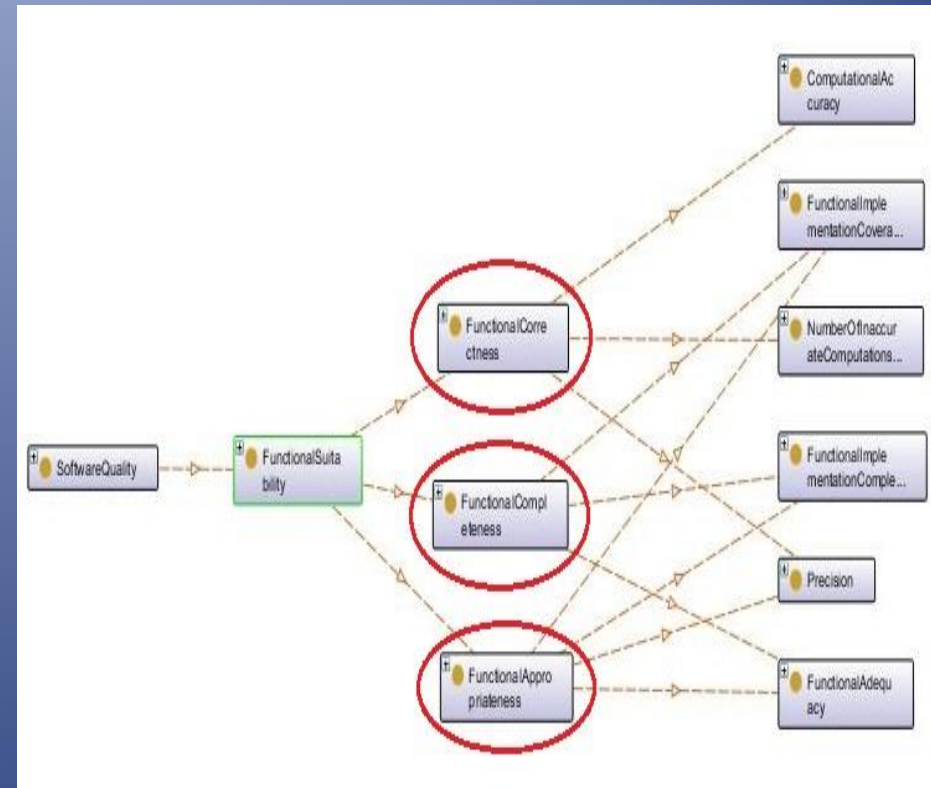
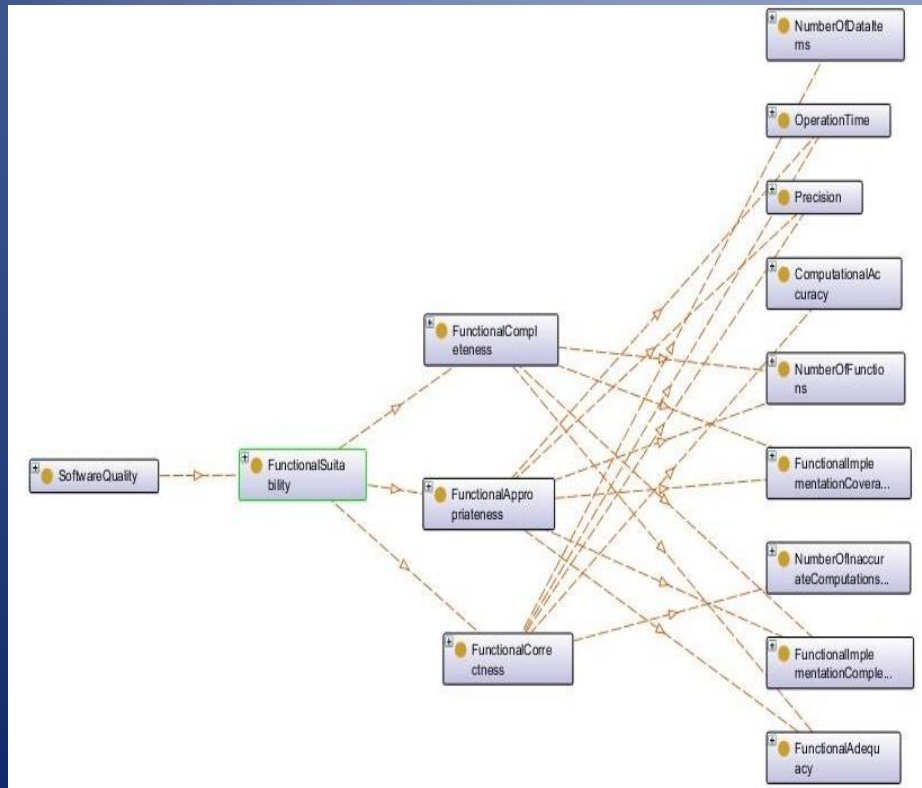
7

- If there are subcharacteristics and characteristics, values of which cannot be determined on the basis of measures (from the SRS), then complement of this specification by the necessary measures

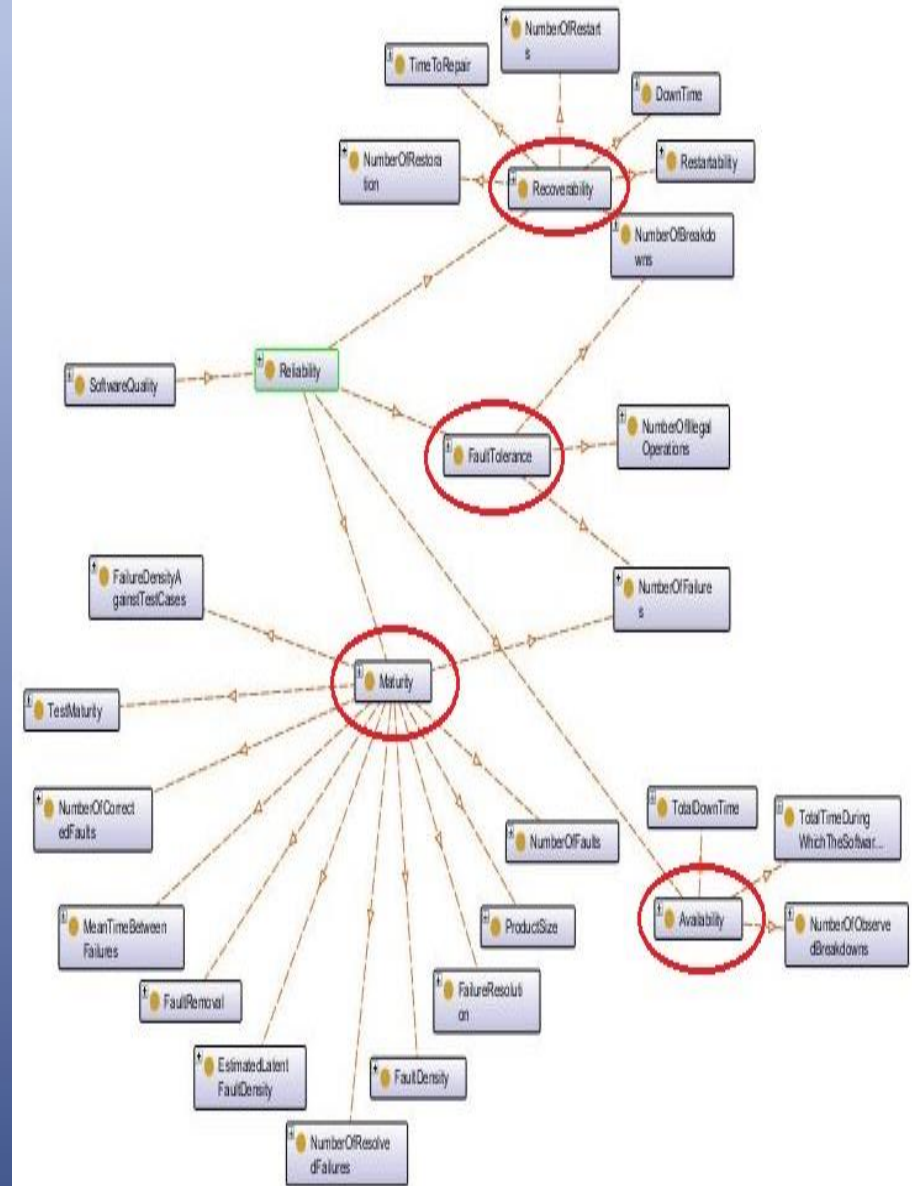
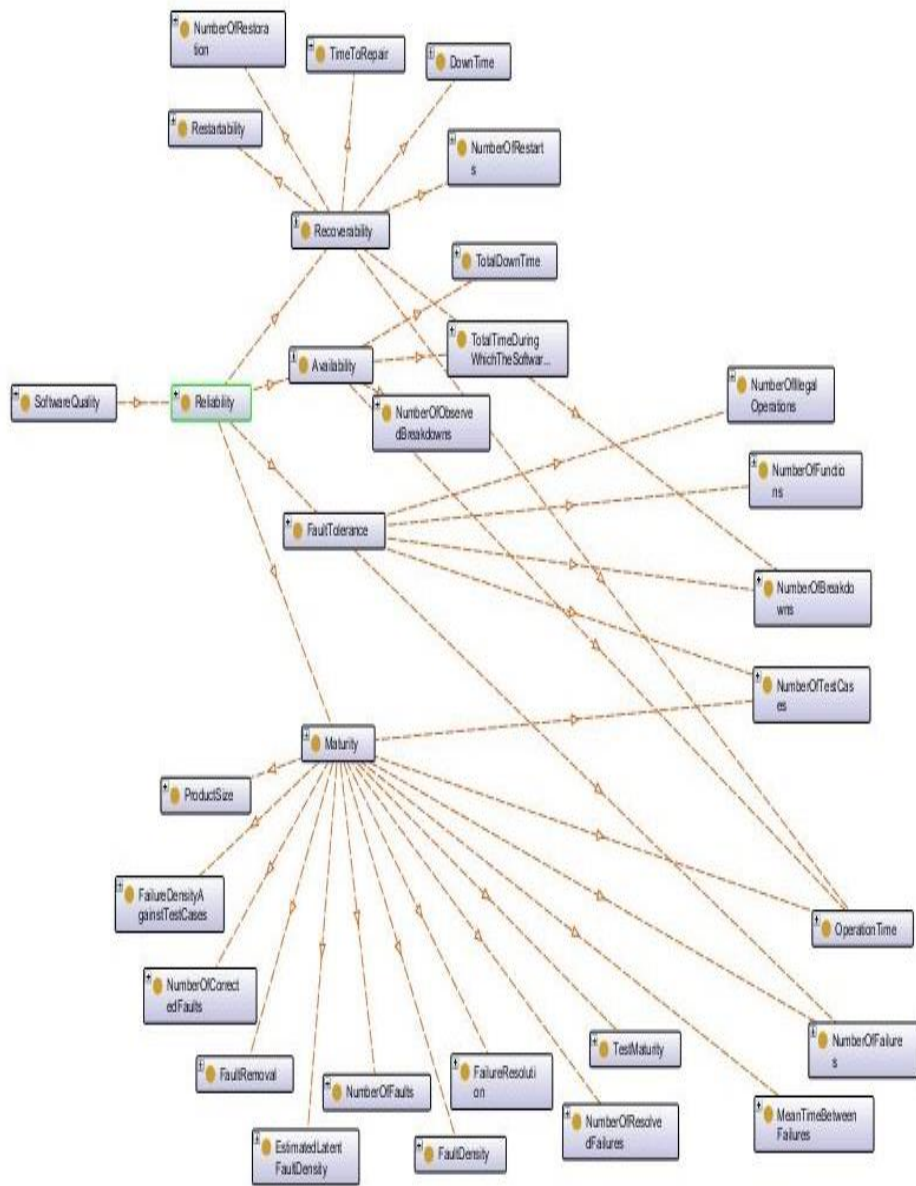
8

- Repeating the steps 2-7 until all quality characteristics and subcharacteristics will be possible to identify or until the conclusion will be formed, that data for software quality determination are insufficient

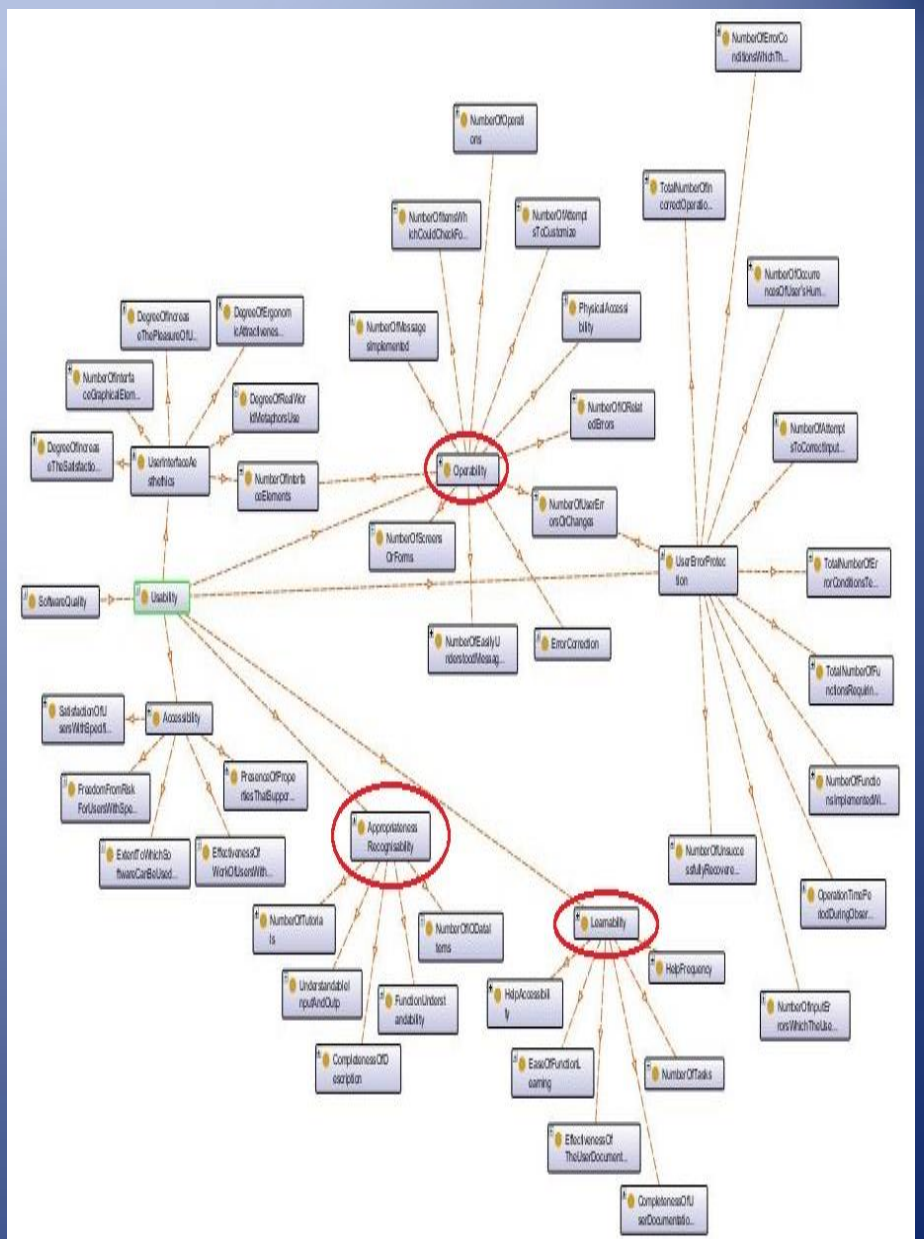
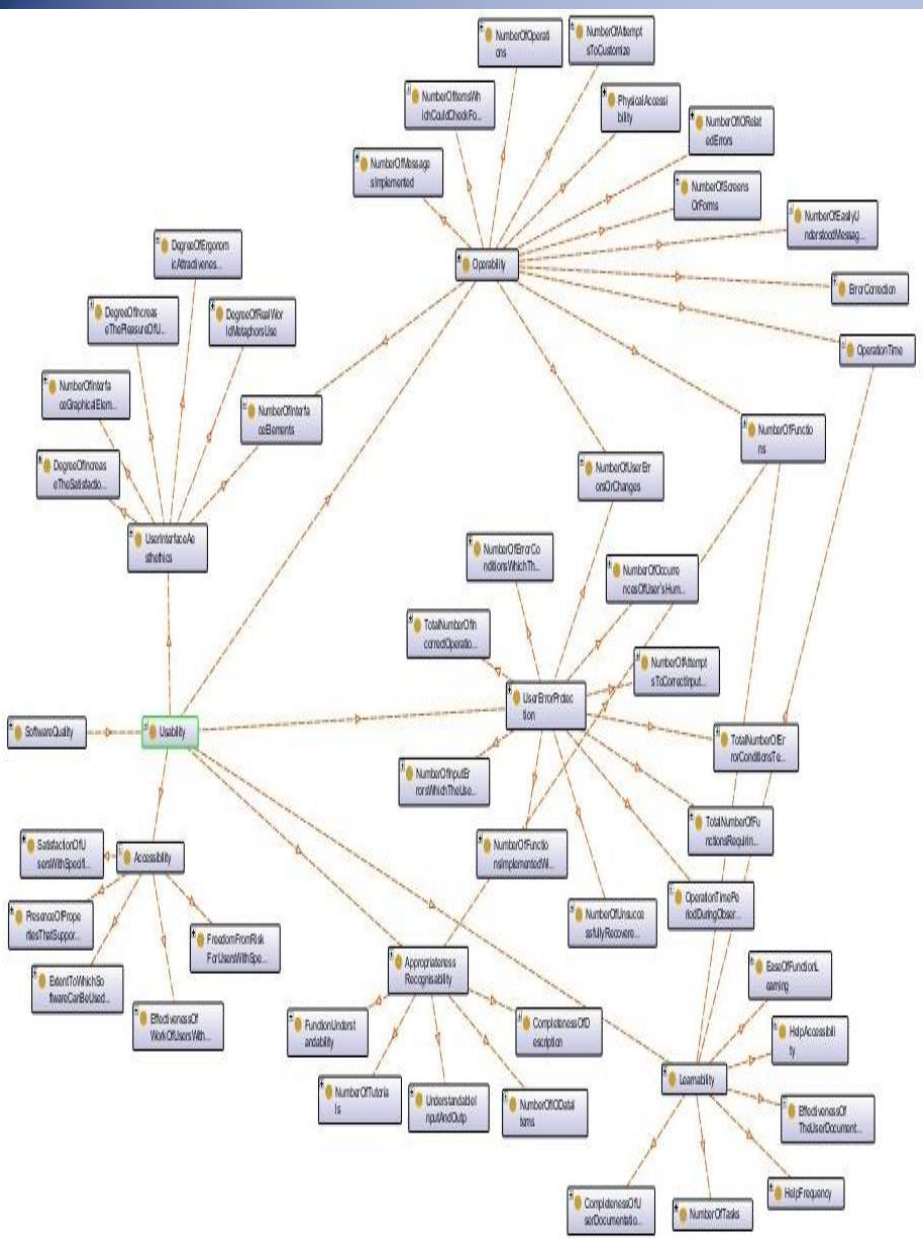
Experiments: Assessment of Information Sufficiency for Determination of Quality of Software of Automated System for Large-Format Photo Print



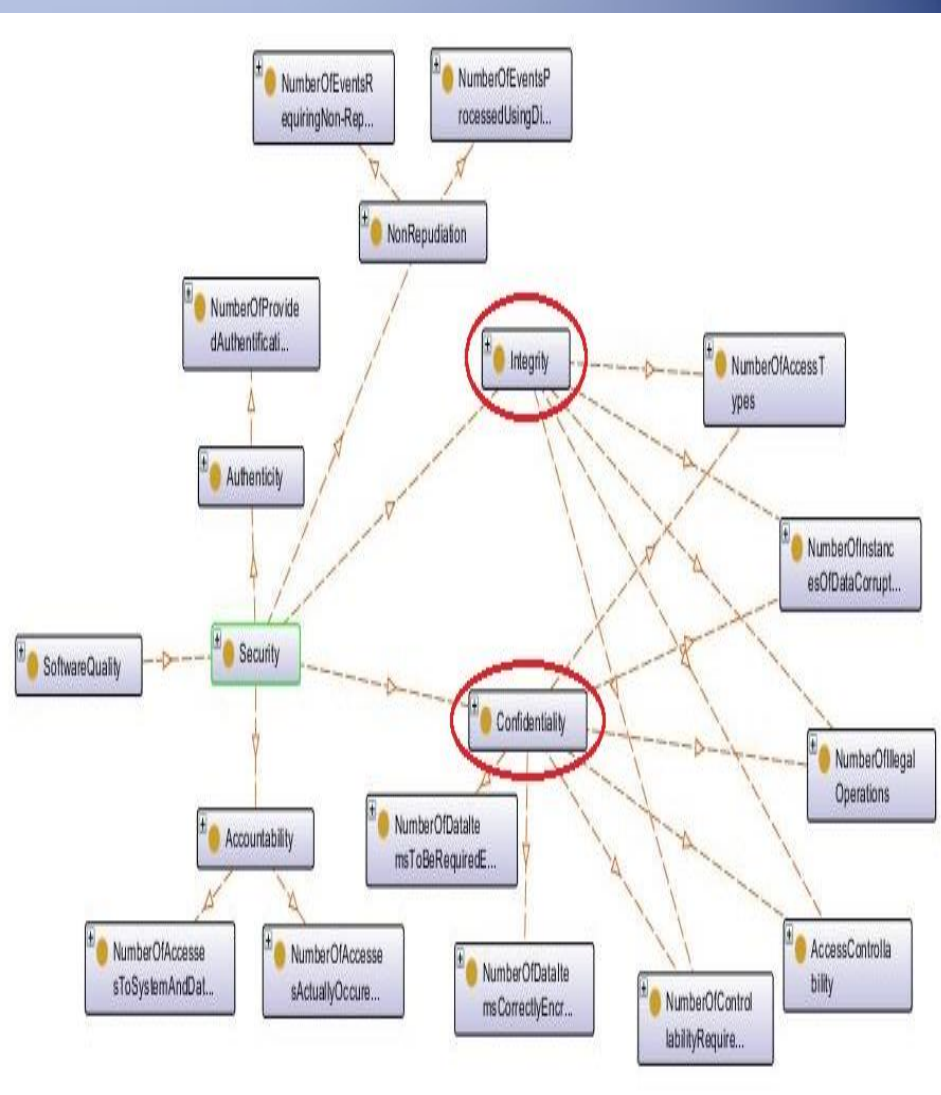
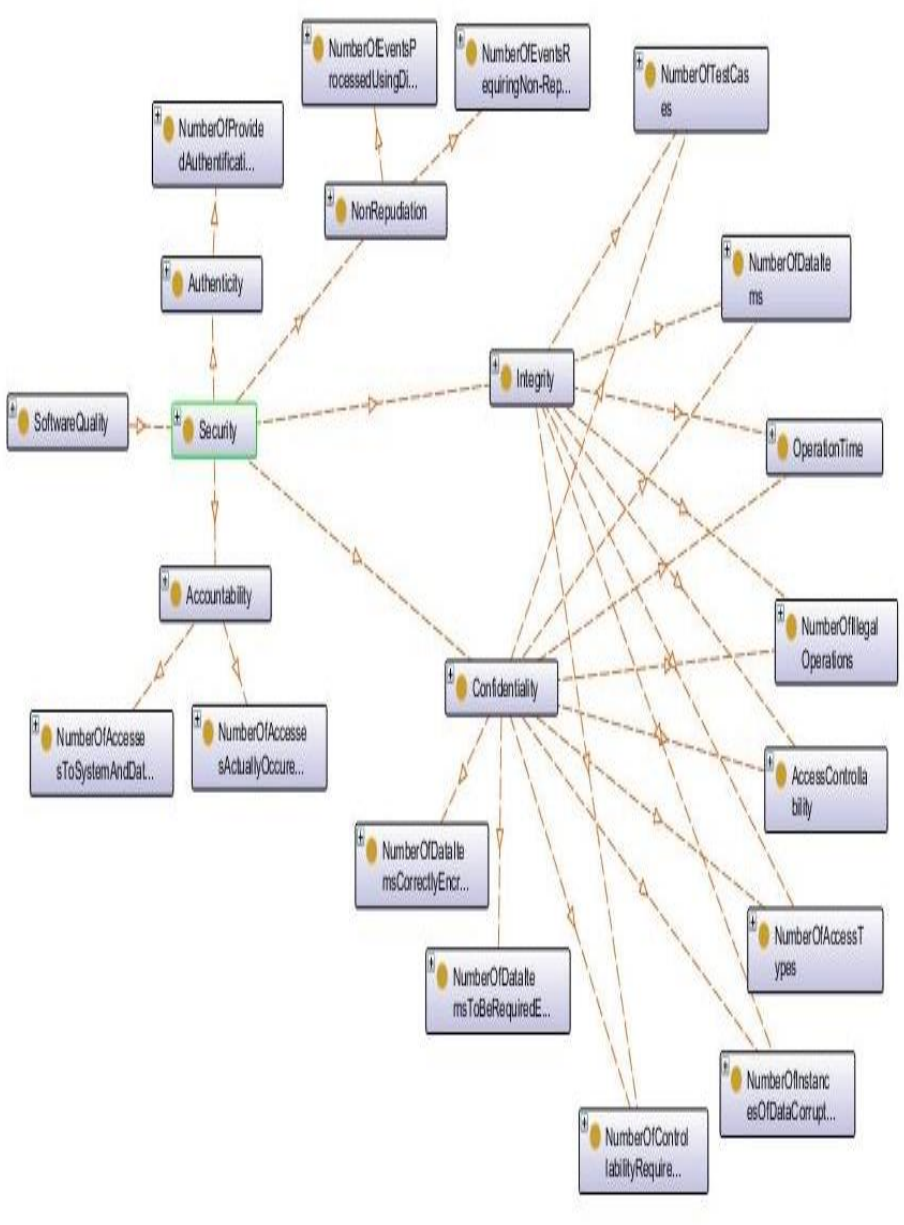
Base ontology and ontology for concrete software project for Functional Suitability



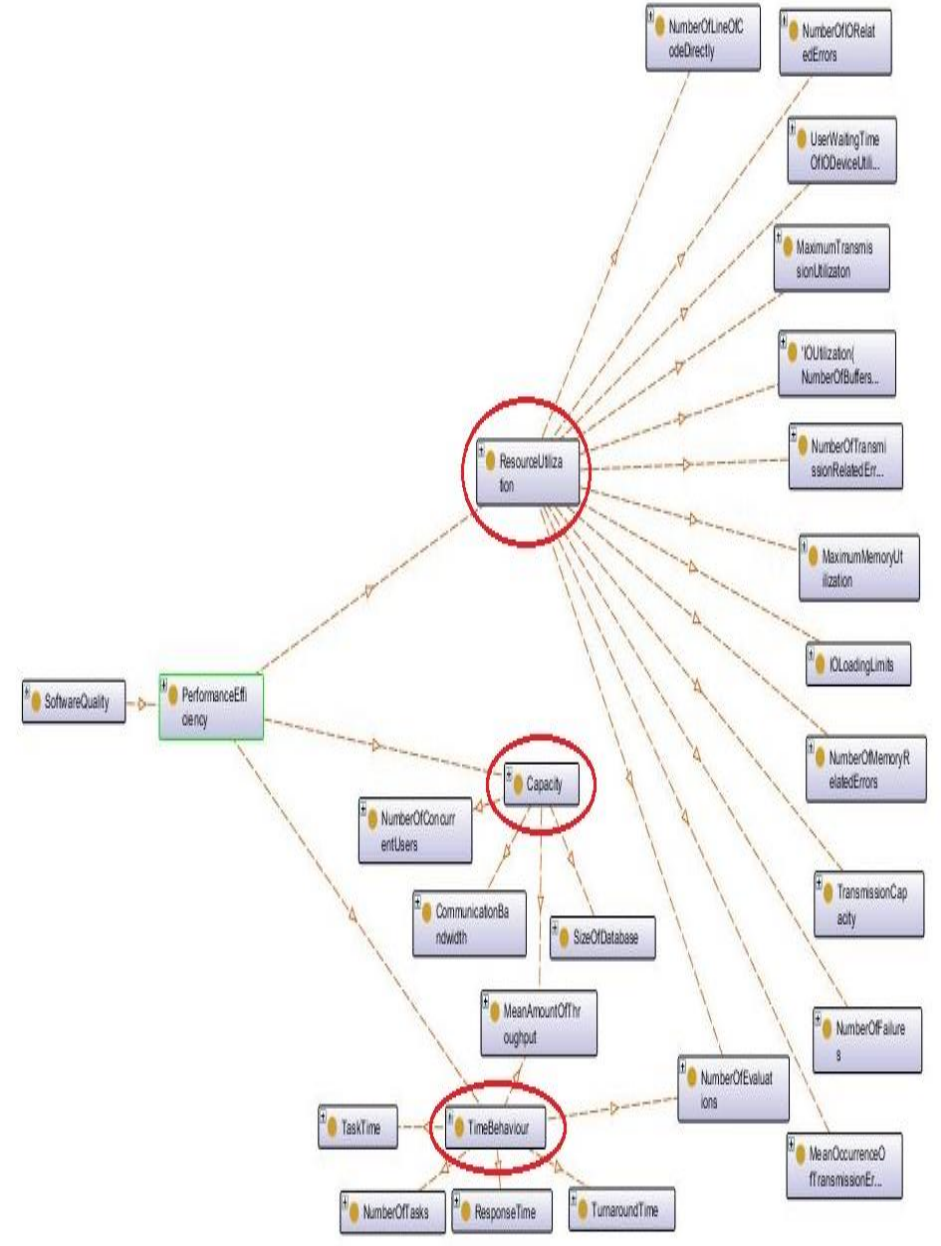
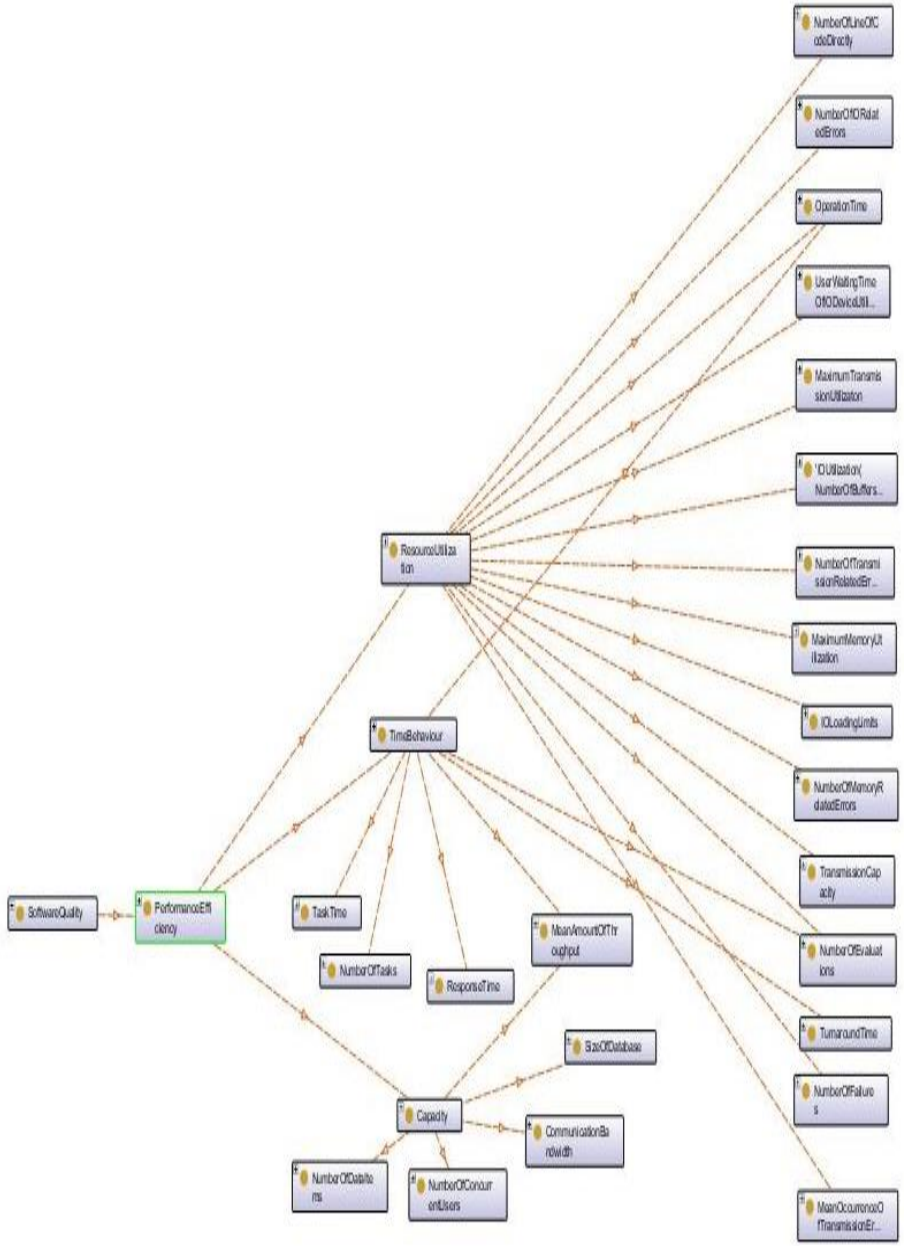
Base ontology and ontology for concrete software project for Reliability



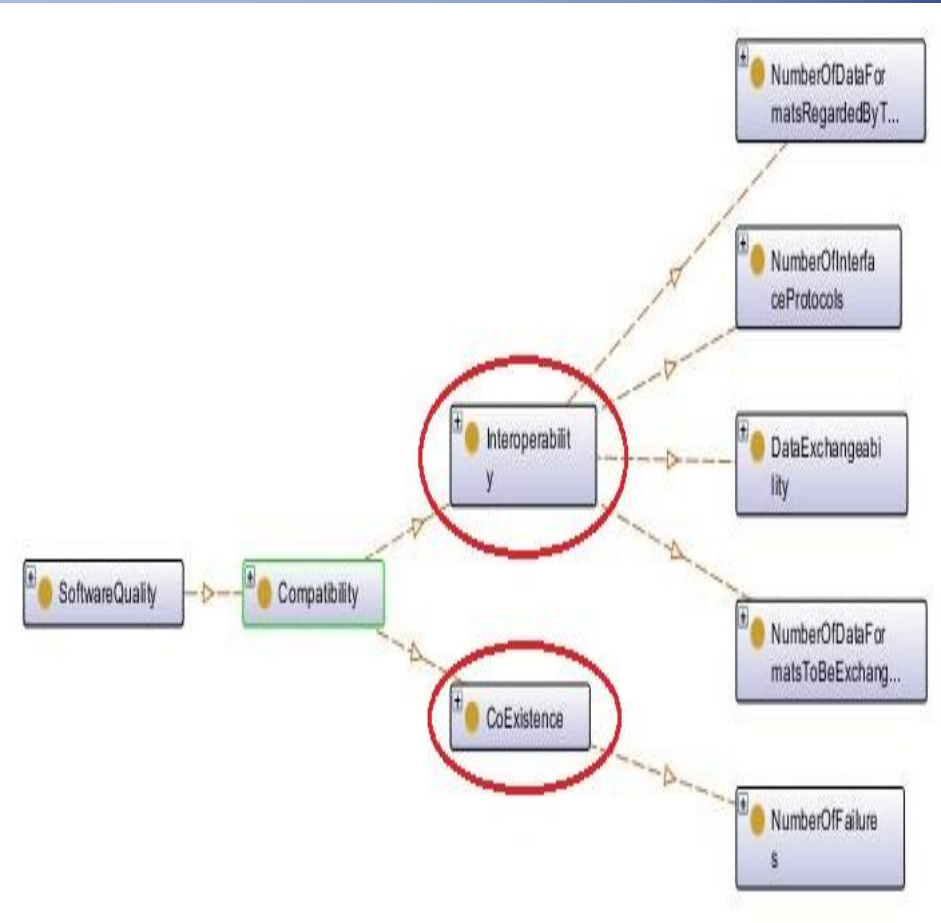
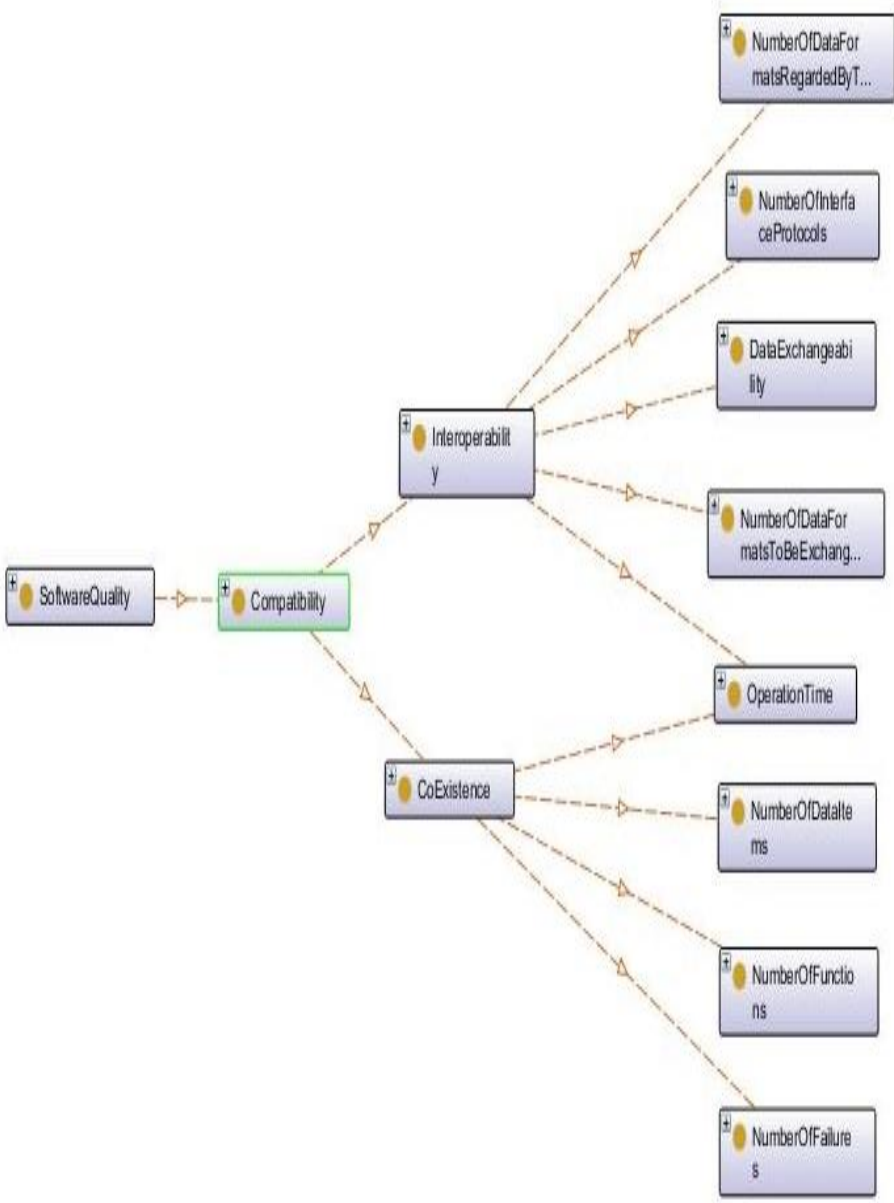
Base ontology and ontology for concrete software project for Usability



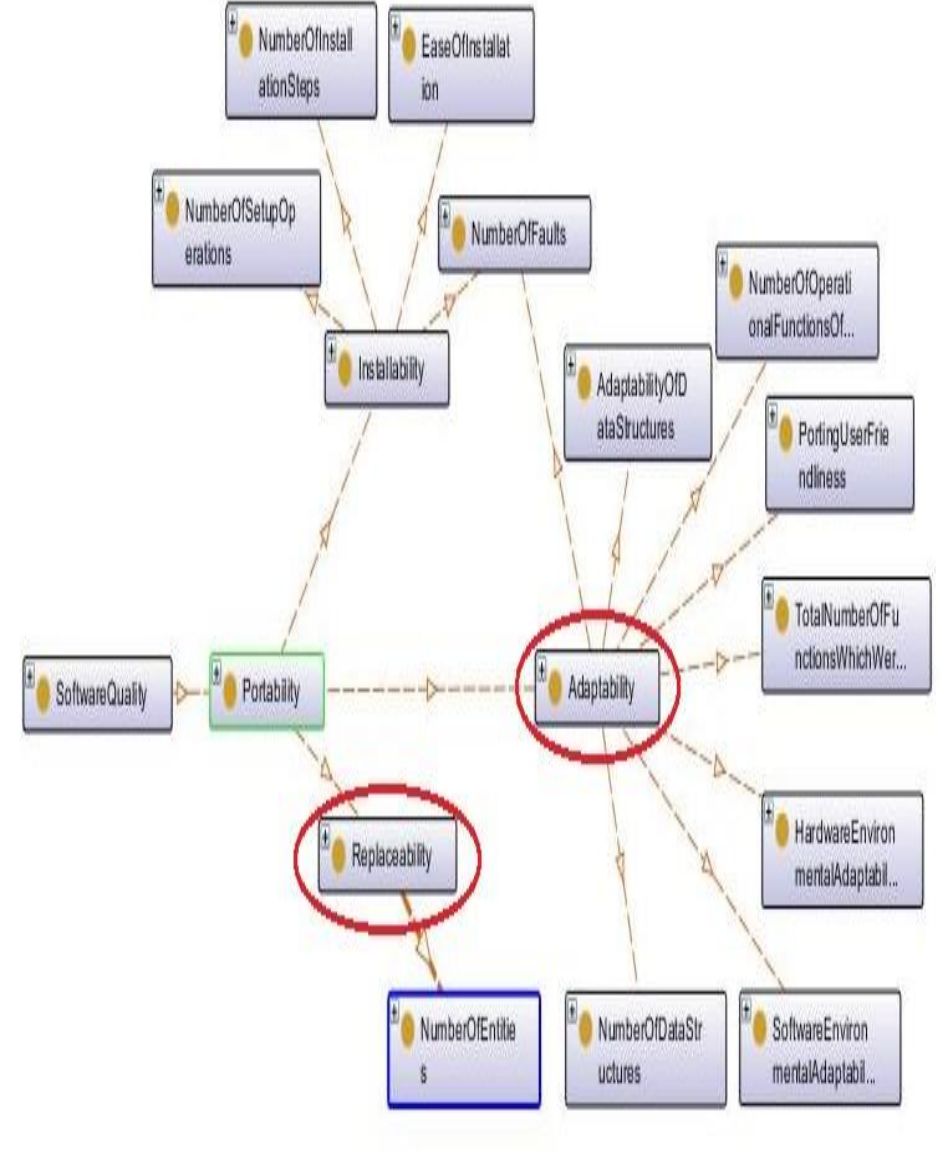
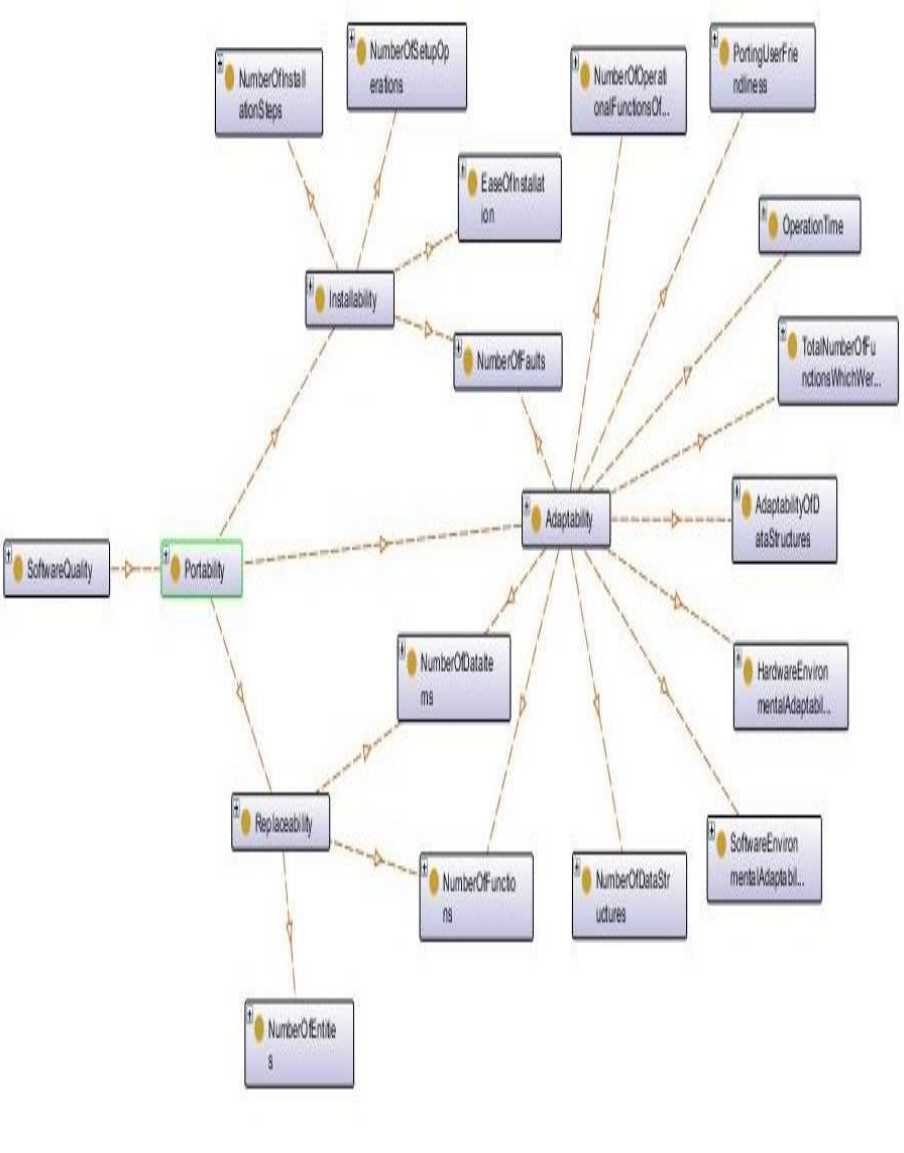
Base ontology and ontology for concrete software project for Security



Base ontology and ontology for concrete software project for Performance Efficiency



Base ontology and ontology for concrete software project for Compatibility

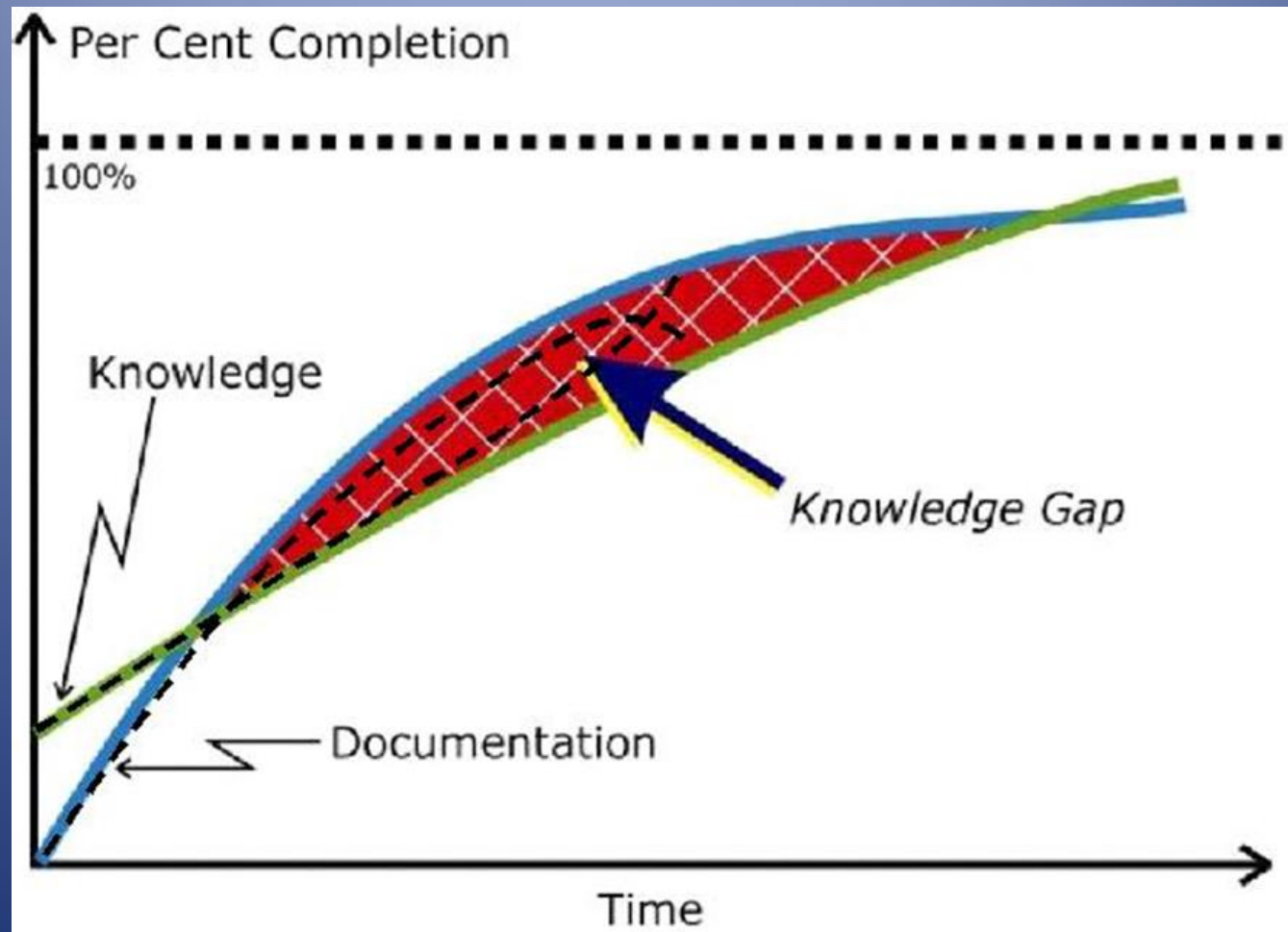


Base ontology and ontology for concrete software project for Portability

Results

- Lack of 4 measures in SRS (Number of Functions, Number of Data Items, Operation Time, Number of Test Cases) – **impossibility of determination of all quality characteristics**
- Complement of the SRS by 2 measures (Number of Functions, Number of Data Items) – **impossibility of determination of all software quality characteristics (still insufficient information).**

- But!!! We have changes in Functional Completeness, Capacity, Appropriateness, Recognisability, Analyzability, and Replaceability – **REDUCING THE SIZE OF KNOWLEDGE GAP**



Conclusions

- The measures analysis is an effective mean of assessing the software quality upon availability of veracity information for it conduct. One of the factors affecting the veracity of such information is sufficiency of the volumes of information about measures in the SRS. Therefore, **solving the task of assessment of sufficiency information about measures in the SRS generally enhances the veracity of software quality assessment.**
- In the analysis of software quality subcharacteristics (as sources of information) the cross-correlation of these subcharacteristics because they have joint measures. **Correlation of subcharacteristics, that displayed by base ontology, should be considered because it can reduce the accuracy and veracity of software quality assessment.**
- For displaying of the knowledge of experienced professionals we selected **ontologies** that **became the basis of the approach to the assessment of information sufficiency for software quality determination** (according to ISO 25010: 2011).
- **Proposed approach provides the conclusion about impossibility of determination of certain software quality subcharacteristics and characteristics, and about necessity of complement of the SRS by required missing attributes - REDUCING THE SIZE OF KNOWLEDGE GAP**

THANK YOU FOR ATTENTION!



Tetiana Hovorushchenko

Doctor, Senior Researcher, Associate of Professor,
Lecturer of System Programming Department
tat_yana@ukr.net , govorushchenko@gmail.com

Oksana Pomorova,

Doctor of Science, Full Professor
Head of System Programming Department
o.pomorova@gmail.com