CAcTUS
Context-Awareness Testing for Ubiquitous Systems

SYSTEMATIC LITERATURE REVIEW PROTOCOL:

INVESTIGATING CONTEXT AWARE SOFTWARE TESTING STRATEGIES

PROJECT REPORT 01/2014

Felyppe Rodrigues da Silva

Santiago Matalonga

Guilherme Horta Travassos

Experimental Software Engineering Group at COPPE/UFRJ

December, 2014

Brazil: COPPE/UFRJand UFC/DC
France: UVHC

CNPq – Universal 14/2013
484380/2013-3

Review History

Date	Version	Description	Author(s)
12/29/2010	0.3	Initial Version of overall template. ISERN 2009 Researchers – discussiongroup: Guilherme H. Travassos (COPPE/UFRJ – Brazil), Oscar Dieste (Universidad Politécnica de Madrid – Spain), Martin Höst (LundUniversity – Sweden), Oscar Pastor (Universidad Politécnica de Valencia – Spain)	Guilherme H. Travassos and Vitor Faria Monteiro, COPPE/UFRJ
	1.0	Evolving the protocol to present a more general perspective accordingly the decisions from Madrid´s meeting (Natalia Juristo, Oscar Dieste, Oscar Pastor, Nelly Condory-Fernandez and Guilherme H. Travassos)	Guilherme H. Travassos, COPPE/UFRJ
	1.0.2	Some adjustments and comments from Nelly	Nelly Condory-Fernandez
12/17/2013	1.1.2	Some adjustments regarding the overall organization and topic of research	Fábio, Felyppe and Wladmir
02/08/2013	1.1.3	Cactus Context: research topic tailoring and configuration	Felyppe and Santiago
07/03/2013	1.1.4	Minor adjustments regarding text correction and criteria	Felyppe
07/07/2014	1.1.4	Text Review and minor adjustments	Guilherme
21/10/2014	1.1.5	Text Review and minor adjustments	Felyppe
07/11/2014	1.2	Search String update, PICO structure adjustments, preliminary results insertion, text review and minor adjustments	Felyppe
15/12/2014	1.3	Results added together with the conclusions	Felyppe
23/12/2014	1.4	Final Version	Felyppe e Santiago
17/01/2014	1.4.1	Applying Guilherme suggestions in the Final Version	Felyppe

Index

1 Main Investigation Scenario

Ubiquitous computing shifts the focus from technology towards the user and her needs. Efforts in ubiquitous computing expand the place and mode of interaction beyond the desktop, into everyday spaces in which the main challenge is to make the systems useful in various situations that may be encountered in the real world(Lei Tang et al, 2011). A key tenant of ubiquitous computing is the concept of invisibility, where technology mustblend into everyday objects (Weiser, 1991). In this context, it is essential that the different devices and services are capable of inter-operate among themselves without the interaction of the user.

With the increasing advancement in the development of ubiquitous applications, important issues related to Human-Computer Interaction (HCI) must be considered, for example: is the system adding value to the end user (Rocha, 2011), are users satisfied when using these types of applications, or whether they really met the expectations of users. These issues are even more relevant when we consider the diversity of existing technologies (e.g., mobile phone, tablets, iPods, etc…), using different types of interaction (e.g., audio, video, text and images) between the user and the machine. In addition to this, the variety of devices poses challenges to the interoperability and the self-managing and self-organizing needs imposed by the ubiquitous environment dynamics (Rocha, 2007). We argue that interactions issuesin ubiquitous systems go beyond the human computer interaction, including the interaction between different devices and systems, which we define as interaction actor-computer (between the actor and the computer), where the actor can be: a human user, another computer or device; or even another system.

In order to achieve this adaptability towards the use of these different technologies, interaction and interoperability, while minimizing the participation of the user, the ubiquitous system must be aware of its context. Software is defined as context aware when it can obtain information of the actor and theenvironments (computational or physical) in order to provide services or better information adapted for the user (Dey, 2001). Therefore, ubiquitous systems must be able to capture the context and useit to adapt its behavior to help the actorsachieve their tasks.

Different authors have explored the problem of assuring the quality of context-aware systems (e.g., Kjeldskov et al. 2006, Fiotakis et al. 2009, O'Neil et al., 2005). In general, these studies have explored testing in the laboratory and in real environment to identify usability problems that allow the improvement of these systems in order to better serve its users. However, to better control this cycle of evaluation and improvement, these tests must be properly designed and implemented, which leads to different research questions: what tests should be performed to ensure the best actor-computer interaction? How to consider the different possible contexts such tests? Are there methods for designing these tests that take into the context into consideration?

2 Research Protocol

This initial systematic review protocol was based on [Biolchini et al., 2005]. To organize and structure the search string, it explores the PICO approach [Pai et al., 2004]. This approach separates the question into four parts:**P**opulation of interest**, I**ntervention or exposure being evaluated, **C**omparison intervention (if applicable) and **O**utcome. Because of the objective of the study (mainly characterization), it will not be possible to apply any comparison. Therefore, we can classify it as *quasi*-systematic literature review [Travassos et al, 2008].

2.1 Question Focus

This study's research objective is to identify the different available context aware methods for testing software systems and to identify coverage levels that each of the identified methods can attain.

2.2 QuestionQualityand Amplitude

- **Problem:** One of the main difficulties in testing context aware systems is to evaluate the coverage level including factors which might not be straightforward derived from user requirements. Therefore, it becomes important to know whether methods have been developed for the generation of test cases concerned with context aware software systems, and to evaluate the coverage levels of these different methods.

- **Question:**
    - **Main Question:**Which are the existing methods for testing context aware systems? What is the coverage obtained by each one of them?

Note: The Word coverage is used in the Intervention as a proxy for Test Effectiveness measurement. Hence, this word does not appear as keyword in the protocol. Initial trials have shown that including it skewed results since other effectiveness measures can be found.

- **Population:** Sensibility to the context.
  - **Keywords:** "context aware" OR "event driven" OR "context driven" OR "context sensitivity" OR "context sensitive" OR "pervasive" OR "ubiquitous" OR "usability" OR “event based” OR “self adaptive” OR “self adapt”.
- **Intervention Control:** Software testing.
  - **Keywords:** "software test design" OR "software test suite" OR "software test" OR "software testing" OR "system test design" OR "system test suite" OR "system test" OR "system testing" OR "middleware test" OR "middleware testing" OR "property based software test" OR "property based software testing" OR "fault detection" OR "failure detection" OR "GUI test" OR "Graphical User Interfaces test".
- **Comparison:** None.
- **Outcome Measure:** Methodology.
  - **Keywords:** "model" OR "metric" OR "guideline" OR "checklist" OR "template" OR "approach" OR "strategy" OR "method" OR "methodology" OR "tool" OR "technique" OR "heuristics".
- **Excluded Keywords:**
  - **Agent Systems / Multi Agent Systems:** Returned too many articles from other areas.
  - **Combinations of “system” and “software” with “fault detection” and “failure detection”:** Returned fewer results than the actual search string.
  - **Defect detection / error detection:** Returned no useful results.
  - **Cyber Physical:**Returned no useful results.
  - **Words with “-“:** The search machines returned the keywords with “-” as two separated words, so we started using the words in that way.
- **Effect:** Performance measurement of each context aware test methodidentified.
- **Application:** Support test planningor design.
- **Experimental Design:** None statistical method is going to be applied.
- **Control Articles:** There were no control articles for this protocol since this is the first trial of this research.

2.3 SourceSelection

2.3.1 Sources Selection Criteria Definition

- Works presented as articles should be available on the web.

2.3.2 Studies Language

English.

2.3.3 Source Identification

- **Source Search Method:** Search through web search engines.
  - **Main question:** Which are the existing methods for testing context aware systems?
  - **Alternative Question:**What is the coverage obtained by each one of them?

Search Engines:

Table 1: Search engines.

Name	Link
Scopus	http://www.scopus.com/
Web of Science	http://www.isiknowledge.com/
IeeeXplore	http://ieeexplore.ieee.org/

- **Scopus Search String:**
  - TITLE-ABS-KEY(("context aware" OR "event driven" OR "context driven" OR "context sensitivity" OR "context sensitive" OR "pervasive" OR "ubiquitous" OR "usability" OR “event based” OR “self adaptive” OR “self adapt”) AND ("software test design" OR "software test suite" OR "software test" OR "software testing" OR "system test design" OR "system test suite" OR "system test" OR "system testing" OR "middleware test" OR "middleware testing" OR "property based software test" OR "property based software testing" OR "fault detection" OR "failure detection" OR "GUI test" OR "Graphical User Interfaces test") AND ("model" OR "metric" OR "guideline" OR "checklist" OR "template" OR "approach" OR "strategy" OR "method" OR "methodology" OR "tool" OR "technique" OR "heuristics"))
- **Web of Science Search String:**
  - TOPIC: ("context aware" OR "event driven" OR "context driven" OR "context sensitivity" OR "context sensitive" OR "pervasive" OR "ubiquitous" OR "usability" OR “event based” OR “self adaptive” OR “self adapt”) AND TOPIC:("software test design" OR "software test suite"

OR "software test" OR "software testing" OR "system test design" OR "system test suite" OR "system test" OR "system testing" OR "middleware test" OR "middleware testing" OR "property based software test" OR "property based software testing" OR "fault detection" OR "failure detection" OR "GUI test" OR "Graphical User Interfaces test") AND TOPIC: ("model" OR "metric" OR "guideline" OR "checklist" OR "template" OR "approach" OR "strategy" OR "method" OR "methodology" OR "tool" OR "technique" OR "heuristics"). Timespan: All years. Search language=English

- **IEEEXplore Search String:**
    - (("context aware" OR "event driven" OR "context driven" OR "context sensitivity" OR "context sensitive" OR "pervasive" OR "ubiquitous" OR "usability" OR “event based” OR “self adaptive” OR “self adapt”) AND("software test design" OR "software test suite" OR "software test" OR "software testing" OR "system test design" OR "system test suite" OR "system test" OR "system testing" OR "middleware test" OR "middleware testing" OR "property based software test" OR "property based software testing" OR "fault detection" OR "failure detection" OR "GUI test" OR "Graphical User Interfaces test") AND ("model" OR "metric" OR "guideline" OR "checklist" OR "template" OR "approach" OR "strategy" OR "method" OR "methodology" OR "tool" OR "technique" OR "heuristics"))

2.4 Studies Selection

2.4.1 StudiesDefinition

- **Studies Inclusion and Exclusion Criteria Definition:**

Inclusion Criteria:

- To talk about test strategies; or
- To talk about test design; or
- To talk about test methods; or
- To talk about test metrics; or
- Totalkabouttestingmeasurement; or
- To talk about fault detection; or

- To talk about error detection; AND
- To present characteristics of context in context-aware software systems; or
- To present some characterization of context in context-aware software systems; or
- To analyze specific problems in sensing variables of context in either:
  - Human Computing Interaction
  - Software Systems'usability

Exclusion Criteria:

- Not talk about test strategies; and
- Not talk about test design; and
- Not talk about test methods; and
- Nottalk about test metrics; and
- Nottalkabouttestingmeasurement; and
- Nottalk about fault detection; and
- Nottalk about error detection; OR
- Notpresent characteristics of context in context-aware software systems; and
- Notpresent some characterization of context in context-aware software systems; and
- Notanalyze specific problems in sensing variables of context in either:
  - Human Computing Interaction
  - Software Systems'usability; and
- Being older than 2000;

- **Study type definition:** Articles related to context-aware systems testing.
- **Procedures for Studies Selection:** Read the title and the abstract of the each retrieved study and evaluate it according to inclusion and exclusion criteria.
- **Acceptance Criteria:** Three distinct readers will evaluate each study. The studies acceptance criteria will happen as follows:
  - **All three readers accept:** The study is included.
  - **Two readers accept and one is in doubt:**The study is included.
  - **One reader accepts and two are in doubt:** The study will bediscussed in group.
  - **Two readers accept or are in doubt and one reader exclude:** The study will be discussed in group.
  - **Two or three readers excluded:** The study is not included.

2.5 Pre-Execution / Execution Results

The Final Extraction from the database was carried out on 30thOctober 2014. Below is shown the pre-execution trials followed by Table 2, shows the references retrieved by each search engine:

- **Trial 1:** Event Driven
    - **Process:**Use only the keyword "Event Driven" in the Population group of the PICO approach and see if the keyword is useful for the research.
    - **Date of execution:** 1st April 2014
    - **Number of Articles Found:** 52
    - **Number of Articles Selected:** 11
    - **Conclusion:**
        - Keep the keyword "Event Driven" in the Population Group.
        - Remove "-" from the keywords.
        - Better alignment of the readers perspective
        - Calibration of the Quality Evaluation Form and the Data Extraction Form

- **Trial 2:** Fault/Error/Failure/Defect Detection
    - **Process:**Use only the keywords "Fault Detection", "Error Detection", "Failure Detection" and "Defect Detection" in the InterventionGroup of the PICO approach and see if the keywords are useful for the research.
    - **Date of execution:** 14th May 2014
    - **Number of Articles Found:** 418
    - **Number of Articles Selected:** 24
    - **Conclusion:**
        - Remove the keywords "Error Detection" and "Defect Detection" from the Intervention Group.
        - Remove combinations of "system" and "software" with "fault detection" and "failure detection", because they returned fewer results than the final search string.
        - Inclusion of other keywords in the search string found during the articles reading, some synonyms of existing keywords and others related to Graphical User Interface.
        - Calibration of the Quality Evaluation Form (regarding test techniques) and the Data Extraction Form using a few taxonomies (domain, application and software category).

- **Trial 3:** Execution
  - **Process:** Use the full search string.
  - **Date of execution:** 29th May 2014
  - **Number of Articles Found:**1121
  - **Number of Articles Selected:**54
  - **Conclusion:**
    - Include the keywords “event based”, “self adapt”, “self adaptive” and “cyber physical” and re-execute the search string.
    - Find a way to classify the studies to facilitate the comparison between them (ISO 29119).

- **Trial 4:** Re-Execution
  - **Process:** Use the full search string with the added keywords from the execution conclusion.
  - **Date of execution:** 30th October 2014
  - **Number of Articles Found:** 1820
  - **Number of Articles Selected:** 21
  - **Conclusion:**
    - Remove the keyword “cyber physical”
    - Update the selected articles group, once some articles are updated versions of others previously selected and some others were just selected more than one time.

Table 2: Search Engines Final Results

SearchEngine	Numberofarticlesfound
Scopus	816
Web of Science	224
IEEEXplore	780
Total	**1820**

2.6 Information Extraction Strategy

For each selected paper the following information shall be extracted and managed using the JabRef reference tool (http://jabref.sourceforge.net/):

Table 3: Information extraction fields

Field	Description
Title	The title of the paper
Authors	List of authors, including email addresses and affiliation
Year of Publication	The year the paper was published
Source of Publication	Name of the Journal, Conference or place where the paper was published
Abstract	The complete abstract of the paper
Context awareness variables	The list of context awareness variables mentioned in the article of the context of the software system,
Context variables sensor method	Context variables must be sensed in order to provide meaningful information to the user. Describe here the method.
Independent variables	Independent variables that were included in the study
Dependent variables	Dependent variables that were included in the study
Test strategy	Name or list of test strategy used to verify the software system.
Domain type were the proposal has been applied	Application Domain according to Kotonya et al.´s taxonomy see Annex I - Types of system Taxonomy
Application type were the proposal has been applied	Application type according to Kotonya et al.´s taxonomy see Annex I - Types of system Taxonomy
Software Systems category	Software Category according to Pressman 2010. See Anex
Type of experimental study	Types of empirical studies according to Wynekoop and Conger with definitions taken from ESE Wiki.

2.7 Paper Quality Evaluation Criteria

The following criteria will be used to evaluate the quality of the selected papers. It aims to highlight those papers that could be more related with the investigation theme and, consequently, giving more confidence on the final result.

- **Criteria related to testing strategy:**
  1. Is the testing strategy formally described (using a formal language, SPEM, UML) (1 pt)
  2. Does the testing strategy include context indicators within the proposal (1 pt)?
  3. Was it possible to extract all data proposed in section 2.6? (0.5 per column starting at context variables)

- **Criteria related to test case generation:**
  4. Is there any description about how the test cases have been derived? (1 pt)
  5. Is there any description about restrictions and conditions about the applicability of the proposal? (1 pt)
  6. Is it possible to identify for which types of system can the proposal be used? (0,5 pts for each system type)
- **Criteria related to the proposal background theory or applicability:**
  7. Does the paper describe any adaptation/evolution of pre-existent approach? (1 pt)
- **Criteria related to the proposal evaluation:**

  Note: If the paper describes a normative writing or Basic Research it takes 0 for this sections (i.e. the proposal has not been evaluated in the field, or simulation studies).

  8. Is there any description about measuring or evaluation of a testing design strategy)? (1 pt)
  9. Does the article describe the application of the proposal in different settings? (0,5 x number of settings)

  Note: For instance of a Case Study research, complemented by another type of study (experiment, simulation, survey) counts as 1 (0,5 + 0,5)

  10. Are context variables identified in the evaluation of the study? (0,5 pts for each variable)

  Note: Question is aimed at the context variables that have been empirically evaluated. Those present in a Research Background section or similar are not counted.

  11. Are test cases/strategies designed for those context variables? (0,5 pts for each variable)

3 Results

Table 4: Total of articles at each stage of the review.

Stage	Numberofarticles
Articles found	1820
Duplicates	142
Articles selected by the inclusion criteria	110
Articles selected after meeting of the reviewers	75
Articles kept after full reading	11

3.1 Articles kept after full reading

1. Alsos, O. and Dahl, Y. (2008). Toward a best practice for laboratory-based usability evaluations. In: ACM International Conference Proceeding Series. pp.3-12.
2. Amalfitano, D., Fasolino, A., Tramontana, P. and Amatucci, (2013). Considering context events in event-based testing of mobile applications. In: Proceedings - IEEE 6th International Conference on Software Testing. pp.126-133.
3. Bo, J., Xiang, L. and Xiaopeng, G. (2007). MobileTest: A Tool Supporting Automatic Black Box Test for Software. pp.8-8.
4. Canfora, G., Mercaldo, F., Visaggio, C. and D'Angelo, (2013). A case study of automating user experience-oriented performance testing. In: Proceedings - IEEE 6th International Conference on Software Testing. pp.66-69.
5. Merdes, M., Malaka, R., Suliman, D. and Paech, B. (2006). Ubiquitous RATs: How resource-aware run-time tests can improve ubiquitous. In: SEM 2006: Sixth International Workshop on Software Engineering and. pp.55-62.
6. Ryan, C. and Gonsalves, A. (2005). The effect of context and application type on mobile usability: An. In: Conferences in Research and Practice in Information Technology Series. pp.115-124.
7. Satoh, I. (2003). Software Testing for Ubiquitous Computing Devices. In: IASTED International Multi-Conference on Applied Informatics. pp.553-558.
8. Tse, T. and Yau, S. (2004). Testing context-sensitive middleware-based software applications. pp.458-466.
9. Wang, H., Zhai, K. and Tse, T. (2010). Correlating context-awareness and mutation analysis for pervasive. In: Proceedings - International Conference on Quality Software. pp.151-160.
10. Wang, Z., Elbaum, S. and Rosenblum, D. (2007). Automated Generation of Context-Aware Tests. pp.406-415.
11. Wang, H., Chan, W. and Tse, T. (2014). Improving the Effectiveness of Testing Pervasive Software via Context Diversity. ACM Transactions on Autonomous and Adaptive Systems, 9, pp.1-28.

3.2 Quality Evaluation Analysis

Table 5: Paper quality evaluation results.

Paper	Question 1	Question 2	Question 3	Question 4	Question 5	Question 6	Question 7	Question 8	Question 9	Question 10	Question 11	TOTAL
Alsos2008	0	1	4,5	0	1	0,5	0	1	0	2,5	2,5	**13**
Amalfitano2013	0	1	4,5	1	0	0,5	1	1	1	4	4	**18**
Bo2007	0	1	4,5	1	0	0,5	0	1	0	0,5	0,5	**9**
Canfora201366	0	1	4,5	0	0	0,5	1	1	1	1	1	**11**
Merdes2006	1	1	4	0	0	0	0	1	0	0,5	0	**7,5**
Ryan2005	0	1	3,5	1	0	0	0	1	0	1	0,5	**8**
Satoh2003	0	1	3	0	0	0	1	0	1	1	1	**8**
Tse2004	1	1	3,5	1	0	0	0	0	0	1	1	**8,5**
Wang2007b	1	1	4,5	1	0	0	0	1	0,5	2	2	**13**
Wang2010a	1	1	3,5	1	0	0	1	1	0	0,5	0,5	**9,5**
Wang2014	1	1	4	1	1	0	1	1	1,5	0,5	0	**12**

- Just half of the selected articles formally describe their approach.
- All of the selected articles shows an approach related with context indicators.
- Although being possible to classify the articles regarding the selected taxonomy, just 5 out of 12 could be classified completely (in aspect of domain, application, type of study, etc)
- At least 66% of the approaches show how the test cases were derived.
- Less than 20% of the approaches show threats to validity, restrictions or limitations, putting the study reliability at risk.
- More than 60% of the articles do not describe for which kind of system the approaches can be applied, putting the study applicability at risk.
- More than half of the studies are completely new approaches, not evolved from any other existing method.
- More than 80% of the papers were evaluated somehow or have some evidence of their behavior.
- Less than half of the approaches were evaluated using more than one different configuration, implying less of confidence in its behavior in other configurations.
- In all studies, at least one context variable was used in the evaluation process.

- More than 80% of the studies had their evaluation process modeled regarding the context variables.

3.3 DataExtraction

In this section we present the data extracted from the 11 articles according to the section 2.6. Data considered inconclusive were found not to be relevant in some part of the research were not included.

Table 6: Context Variables.

ARTICLE	CONTEXT VARIABLES
Alsos2008	Pacient satisfaction
	Doctor Satisfaction
	Required user attention
	Predictability of system behavior
	Integration with the work situation
Amalfitano2013	GPS signal
	Network signal
	Incoming call
	incoming sms
	Screen rotation Battery level
	USB plugging
	Minimize the app
Bo2007	Test type: * Function test (3g, wifi, bluetooth, incoming calls, sms...) * Volume test (quantity of information such as emails or sms) * Multiple state testing (The phone might be charging and in a low state mode to save power for example) * Multiple task testing (eg. Independent actions together, such as "Receive SMS" and "Enter in charge state") * Boundary testing (eg. Dependent actions together"Incoming Call"and "Clear calls history")
	End users
	Wireless signals
	Other devices
Canfora201366	Human reaction
	UX Score (User satisfaction)
Merdes2006	Other Devices
	Users
	Services of unknown origin
	Implementations
	Conectivity
	Location
	Resource-dependent behaviour
Ryan2005	Location context
	Application type
	Performance
	Error rates
	User satisfaction

	Learnnability
	Efficiency
	Easy of use
	Context awareness
Satoh2003	Network dependency and interoperability
	Mobility and Disconnection
	Spontaneous and Plug-and-Play Management
Tse2004	Time
	GPS Position
Wang2007b	Location
	Battery level
	Time of day
	Environmental readings (e.g. temperature, humidity)
	User preferences (e.g. spoken language, spending limits, ringing profile)
Wang2010a	Location
	Activity information
Wang2014	Context Diversity
	Wifi
	3g
	Bluetooth

Table 7: Dependent Variables.

ARTICLE	DEPENDENT VARIABLES
Alsos2008	Pacient satisfaction
	Doctor Satisfaction
	Required user attention
	Predictability of system behavior
	Integration with the work situation
Amalfitano2013	LOC Coverage
	Method Coverage
Bo2007	Effectiveness
	Efficiency
	Maintenance cost
	Time
	Number of bugs
Canfora201366	Human reaction
	UX Score (User satisfaction)
Merdes2006	
Ryan2005	Performance
	Number of errors
	Learnability
	Efficiency
	Ease of use
	User Satifaction
	Context Awareness
Satoh2003	System Type: UPnP-based Management System X Printer Management System
Tse2004	

Wang2007b	Contextual coverage
	Execution time
Wang2010a	Number of mutants generated
	Ratio of equivalent mutants to all generated mutants
	Ease of killing mutants
	Trend on the number of mutants killed
	Context Diversity
Wang2014	Killed mutants
	Size of candidate test suites
	Faults detection rates (against baseline)
	Coverage

Table 8: Independent Variables.

ARTICLE	INDEPENDENT VARIABLES
Alsos2008	Kind of solution (location-based; token-based)
Amalfitano2013	Testing method: Ripper (Tests GUI events)
	Extended Ripper (Tests also context events)
Bo2007	Test tool (Test Quest Pro; Mobile Test)
	Test type
	Device Type
Canfora201366	Phone speed
	Tool
	User Profile
Merdes2006	
Ryan2005	Location context via GPS simulated
	Application Type (4 implementations Web Based/PC, Web Based, Mobile/Device based PC, Device Based Mobile)
	Use Scenarios (designed)
Satoh2003	Location (network)
Tse2004	
Wang2007b	Tour app (Base, delayShortTourApp, delayLongTourApp)
	Location (registration booth, 2 demo rooms, exit room)
	Visitor’s interests (demos, applications, virtual environment)
	Interest levels (low, middle, high)
	Context-Scenarios (event sequences propagated through the application in the wrong order, Context Dirver Generator Algorithm (3 proposals)
Wang2010a	Location
	Mutation Operatons for Java Programs
Wang2014	Killed mutants
	Size of candidate test suites
	Faults detection rates (against baseline)
	Coverage

Table 9: Studies classification according to section 2.6.

ARTICLE	TYPE OF EXPERIMENTAL STUDY	DOMAIN	APPLICATION	SOFTWARE SYSTEM CATEGORY
Alsos2008	Laboratory Experiment	Healthcare	Emergancy care	Application software
Amalfitano2013	Case Study	Unclassified	Unclassified	Application software
Bo2007	Case Study	Embedded systems	Operating Systems	Embedded software
Canfora201366	Case Study	Telecomunications	Sensors	Application software
Merdes2006	Normative Writing	Utilities	Retaling functions of electrig, water and gas utilities	Application software
Satoh2003	Normative Writing	General Use	Unclassified	Application software
Ryan2005	Case Study	Utilities	Other	System software
Tse2004	Case Study	Utilities	Home care	Unclassified
Wang2007b	Case Study	Utilities	Unclassified	Application software
Wang2010a	Case Study	General Use	Unclassified	Unclassified
Wang2014	Laboratory Experiment	Utilities	Other	Application software

3.4 DataAnalysis

With a view at standardizing Testing Techniques names, it was decided to use the ISO/IEC/IEEE 29119:2013 Software and Systems testing standard in order to classify the observed testing techniques. For each technique the Author had declared to have used, the corresponding ISO 29119 Testing category was assigned to the article, as shown in the following table.

Table 10: ISO 29119:2013 classification.

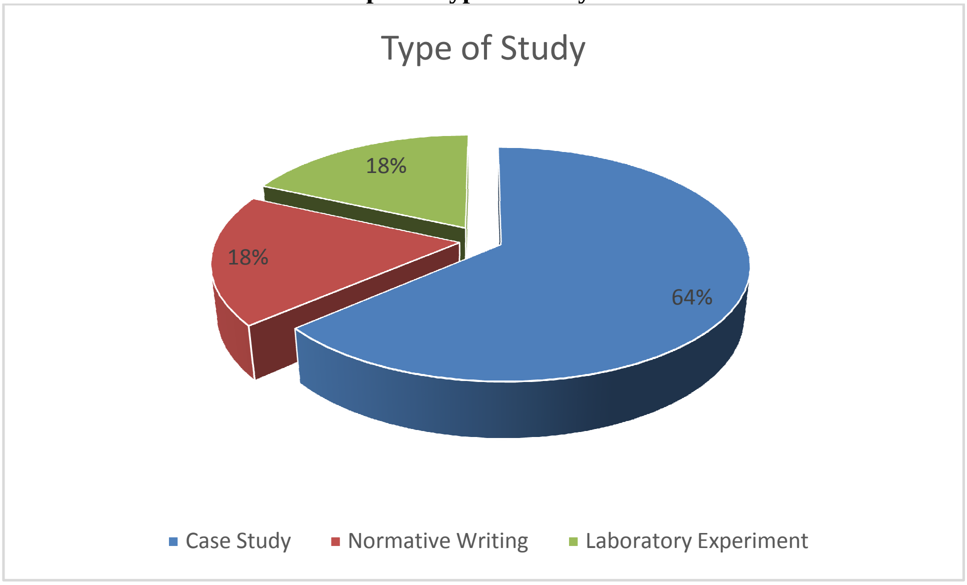
ARTICLE	TEST DESIGN TECHNIQUE (ISO 29119)	TEST TYPE (AUTHOR'S OPINION)	TEST TYPE (ISO/IEC 25010)
Alsos2008	Scenario Testing	Usability Comparative Testing	Usability Testing
Amalfitano2013	Error Guessing	Exploratory Testing	Procedure Testing
Bo2007	Scenario Testing	None	Compatibility Testing
Canfora201366	Scenario Testing	User Experience Testing	Usability Testing
Merdes2006	Scenario Testing	Run-Time testing	Interoperability Testing
Ryan2005	Scenario Testing	Usability Testing	Usability Testing, Functional Testing
Satoh2003	None	Interoperability Testing	Compatibility Testing, Interoperability Testing
Tse2004	Random Testing	Metamorphic testing	Functional Testing
Wang2007b	Scenario Testing	None	Functional Testing
Wang2010a	Syntax Testing	None	Compatibility Testing
Wang2014	Branch Testing	Coverage-based Testing	Procedure Testing

This effort shows that:

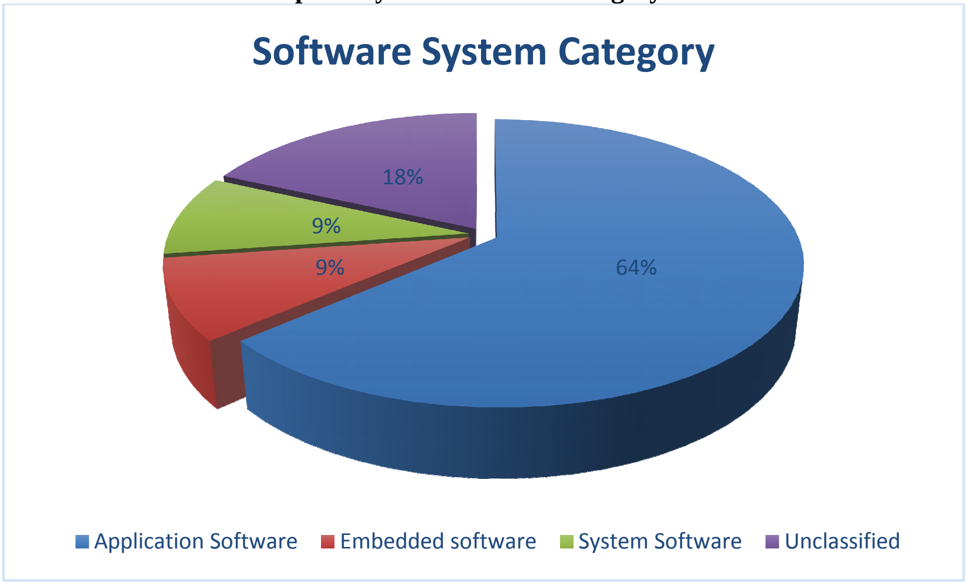
- Hardly ever do the authors and ISO 29119 classifications coincide regarding testing type, showing a lack of consensus over the classification taxonomy.
- The most used Test Design Technique was Scenario Testing (more than 50%). However, scenario testing can hardly show the system perception of context, once its behavior is controlled.

Also regarding the data showed in section 3.3, we could observe the major behavior of Studies Type and Software System Category of the selected papers:

Graph 1: Type of Study.



Graph 2: System Software Category.



With this data we can conclude that most of the proposed approaches aim at solving a very specific problem (application software) without putting it to be tested in a real environment, since most of the studies are laboratory experiments.

3.5 Protocol Questions

- **Main question: Which are the existing methods for testing context aware systems?What is the coverage obtained by each one of them?**

  In spite of the results presented in the previous section, it is our understanding that there is no evidence of specific techniques for context aware software testing.

  Even though the selected studies declare the subject under study to be testing to context aware software systems, the techniques are using conventional (non "context-aware") software testing techniques. With the remarkable exception that are applying these software testing techniques to a context aware software system.

  The notable exception is [She, 2009], where his proposed technique could be used to derive context aware test cases, unfortunately his experimental design does not take care that actual test cases take into consideration context aware variables. And so there is no evidence of actual context aware test cases being produced by the experimental subjects.

  Furthermore, the articles selected can be classified as follows:

  - Proposition and evaluation of proposed frameworks without presenting information about context awareness (showing context variables, but not context awareness testing)
  - Middleware testing
  - Applied testing that does not fit actual classification or expectations (ie. Random or metamorphic testing)

  We can explain this results showing the lack of consensus regarding not only software testing, but mainly context-aware and context in general. Those last two keyword were found being used in a very speculative way, not having much link with out research.

4 Threats to validity

Throughout the execution of the protocol, we took care to reduce the exposure to threats that could invalidate our results. These sections present a discussion of those we identified, mitigate or chose to accept.

4.1 Threat of missing literature

This threat was mitigated by the coverage attained by the selected databases in which the search string was executed. It has already been established by the previous experience of one of the researchers that, at the time this research was carried out, this set of databases provide adequate coverage and minimum overlap. Furthermore, this threat was also mitigated by the iterative nature in which we evolved the search string. Inclusion and exclusion of keyword were always based on trial runs with the keyword on the selected databases. In addition to this, we made no artificial constraint on the subject of the literature retrieved by the search engines. In contrast, the cutoff date on the year 2000 could mean that there is some literature which we might have missed. Nonetheless, we snowballed from the selected literature and found no literature in the references that would pass the inclusion criteria, the results from this process is summarized in the following table.

Table 11: Snowballing results.

Original Source	Snowballed source	Year
Merdes et al.	Wang et al.	1998
Satoh	Abowd et al.	1996
	Schilit et al.	1994

4.2 Threat of selection bias

This threat was mitigated by the process used for the selection of the sources. The criteria that all researchers involved have equal weight in the final decision, and we included an inclusive criteria where all papers which a researcher had doubts were likely to be included.

4.3 Inaccuracy of data extraction

This threat was mitigated by the iterative nature of the execution of the research protocol and the peer review applied by the researchers. Furthermore, the use of inter-rater reliability measures for categorical information on the extraction forms enhances the confidences that the researchers are aligned into the meaning of each extraction.

4.4 Bias on synthesis of information

This threat was mitigated because of the use of already established taxonomies for the aggregation of the data. Extensive effort was put forth in obtaining a shared understanding of the categories in each of the used taxonomies. Needless to say, this shared understanding can differ from the understanding of others. Furthermore, since ISO/IEC/IEEE 29119 part 4 is still in an international draft status, the category used for Test Type can have modifications in the near future.

4.5 Construct validity threat stemming from ISO/IEC/IEEE 29119 Test Type classifications

A construct validity regarding our interpretation and aggregation of data stems from the fact that although we have used the ISO/IEC/IEEE 29119:2013 standard as a reference for software testing taxonomies and vocabulary. During our evaluation and synthesis of information we have come to question the foundation of the definitions in ISO/IEC/IEEE 29119:2013, yet this same standard had been used for information synthesis.

4.6 Construct validity on quality of selected studies

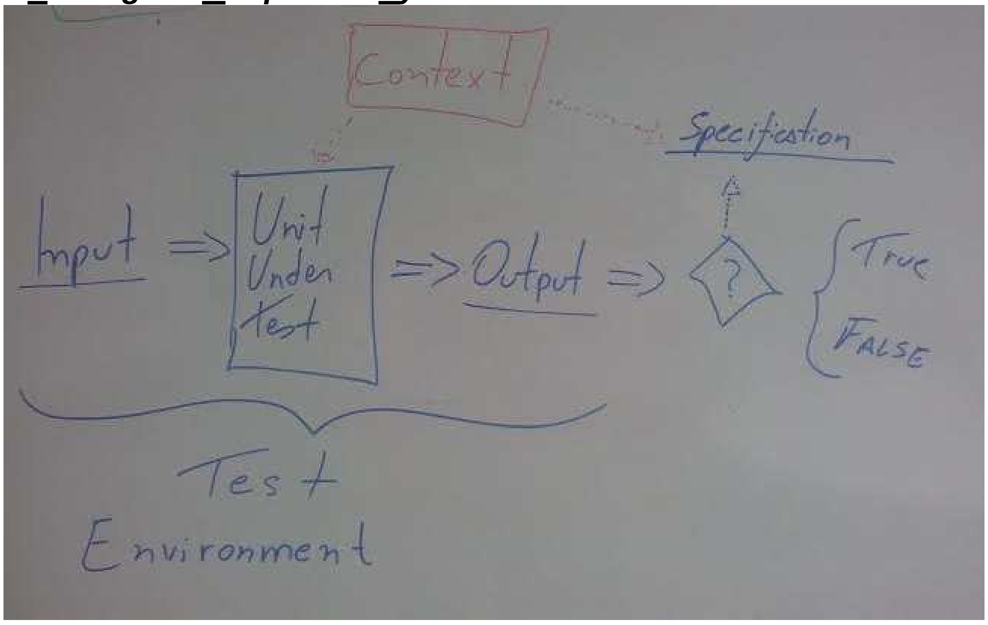
There is a possible source of threat to validity of this study coming from the spread of the quality evaluation points of the selected sources (see Table 5). This could be an indication that the included sources, which were treated as comparable throughout this research, are not comparable and belong to different types of research. At least, studying the selected technical literature, it can be observed that 5 of 11 deal with Usability Testing, that at least can create two different type of sources that could have been treated differently. Nonetheless, given the number of technical literature that the qSLR retrieved, we decided to accept this threat in order to gather more evidence for our study.

5 Conclusions

There seems to be a general lack of consensus regarding some aspects related with our research. Most of the author’s description of software testing and context-aware systems simply does not match. Moreover, most of the found evidence does not regard the context-aware characteristic that we are looking for; it just shows some testing technique in which some context variables are taken into account.

Our perspective of software testing in this research was “Software testing is the procedure of analyzing a software item to detect the differences between existing and required behaviors (as specified in the requirements)”. Having this in mind, we can define contextas“A set of properties that define a possible usage situation for a software system” and context aware as “The expected behavior of a software system after a change occurred in the context”.

This perspective can be observed in the image below, and also, is possible to observe the impact of the context in this perspective.



Context aware systems take into consideration the modification in characteristics concerned with physical structure; environment, user and infrastructure to identify (perceive) context changes (states). Therefore, what was most missed in the found articles was:

- Testing shall consider such context changes, even though the system does not have knowledge about all possible contexts existence.
- What happens when the software (under a context state) is performing its functionality and context`s state changes? This is common situation but not considered in any observed approach.
- What is the expected behavior when the software system is performing and the context state changes? Moreover, if a test case had been applied?
- How to plan testing to capture these variations of context states?
- What is the influence of context changes in overall testing effort?

6 References

Biolchini, J., Mian, P.G., Natali, A.C., Travassos, G.H. (2005). “Systematic Review in Software Engineering: Relevance and Utility”. Technical Report.PESC - COPPE/UFRJ.Brazil.http://www.cos.ufrj.br/uploadfiles/es67905.pdf

Dey, A. K. (2001) “Understanding and Using Context”. Personal and Ubiquitous Computing, v. 5, n. 1, p. 4–7.

Fiotakis, G., Raptis, D., Avouris, N., “Considering Cost in Usability Evaluation of Mobile Applications: Who, Where and When”, INTERACT '09 Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part I, 2009.

Kjeldskov, J., Skov, , M.B., Als , B., Høegh , R., “Is it Worth the Hassle? Exploring the Added Value of Evaluating the Usability of Context-Aware Mobile Systems in the Field”, Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles, 2006

Lei Tang et al, Supporting rapid design and evaluation of pervasive applications: challenges and solutions. In Personal and Ubiquitous Computing Journal, 2011.

O’Neil, E., Klepal, M., Lewis, D., O’Donnell, T., O’Sullivan, D., Pesch, D., “A Testbed for Evaluating Human Interaction with Uniquituous Computing Environments”, Proceedings of the First International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities, pp 60-69, 2005.

Pai, M. McCulloch, M. Gorman, J.D. et al. (2004) “Systematic Reviews and meta-analyses: An illustrated, step-by-step guide”, The National Medical Journal of India, vol. 17, n.2.

Pressman, Roger S. *Software engineering: a practitioner's approach*. New York: McGraw-Hill HigherEducation, 2010. Print.

Rocha, L. S. AdaptiveRME e AspectCompose: Um Middleware Adaptativo e um Processo de Composição Orientado a Aspectos para o Desenvolvimento de Software Móvel e Ubíquo. Dissertação de Mestrado do Programa de Pós Graduação de Ciências da Computação da Universidade Federal do Ceará, 2007

Rocha, L.S.; Bosco Ferreira F, J.; Lima, F.F.P.; Maia, M.E.F.; Viana, W.; de Castro, M.F.; Andrade, R.M.C. Ubiquitous Software Engineering: Achievements, ChallengesandBeyond. Software Engineering (SBES), 2011 25th Brazilian Symposium on 2011.

Travassos, G. H.; Santos, P. M.; Mian, P. G.; Dias Neto, A. C.; Biolchini, J. (2008) "An Environment to Support Large Scale Experimentation in Software Engineering,", 13th IEEE International Conference on Engineering of Complex Computer Systems (iceccs 2008). pp. 193-202DOI: http://doi.ieeecomputersociety.org/10.1109/ICECCS.2008.30

G. Kotonya, I. Sommerville, S. Hall, “Towards A Classification Model for Component-Based Software Engineering Research”. 29th. EUROMICRO Conference, Belek-Antalya, Turkey, IEEE Computer Society, 3-5 September 2003, pp. 43-52

Wynekoop, J.L. and Conger, S.A.: A Review of Computer Aided Software Engineering Research Methods. In Proceedings of the IFIP TC8 WG 8.2 Working Conference on The Information Systems Research Arena of The 90’s, Copenhagen, Denmark (1990)

7 Annex I - Types of system Taxonomy

This Annex describes the classification proposed by G. Kotonya, I. Sommerville, and S. Hall, “Towards a Classification Model for Component-Based Software Engineering Research,” in *EUROMICRO ’03: Proceedings of the 29th Conference on EUROMICRO*, 2003.

This section is an aid to normalize classification criteria among researchers.The left column values can be used to fill the “Domain type were the proposal has been applied”. While the right column must be used to fill in the“Application type were the proposal has been applied”. For more information refer to the aforementioned reference, the classification is transcribed here for practical purposes.

Domain	Application
Avionics	Air Traffic control
	Electronic Warfare
Command and Control	Space
	Satellite
	Other
Embedded systems	Operating Systems
	I/O Controllers
	ASIC
	Other
Electronic Commerce	Agents
	Brokerage
	Electronic Data Interchange
Finance	Accounting
	Banking
	Insurance
Healthcare	Emergency care
	Home care
	Primary care
Real-time	Controllers
	Sensors
	Signal processors
Simulation	Environmental simulations
	War gaming
Telecommunications	Network management
	Network engineering
Utilities	Transmission
	Distribution
	Marketing
	Retailing functions of electric, water and gas utilities

In addition to the previous classifications, for the purposes of data extraction we have added these two new categories:

Domain	Application	Explanation
General Use	N/A	To be used when the author does not specify an application domain
Unclassified	Unclassified	To be used when the researcher is unable to classify the paper

8 Annex II - Types of experimental study Taxonomy

This Annex describes the classification proposed by Kjeldskov, J., & Graham, C. (2003). A Review of Mobile HCI Research Methods.In Human-computer interaction with mobile devices and services. (pp. 317–335). doi:10.1007/978-3-540-45233-1_23

This section is an aid to normalize classification criteria among researchers.The classification to be used for “Type of experimental study” data extraction category (see section Information Extraction Strategy) is defined on page 2of the aforementioned reference, as mentioned, for consistency purposes definitions of each type is taken from the ESE Wiki.

Setting	**Method name (classification)**	**Definition**
Natural Setting	Case Study	A case study is an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between the phenomenon and context are not clearly evident
	Field Study	A Field Study typically examines data from a variety of projects simultaneously. Usually, data is extracted from various sources, and for each activity to determine the effectiveness of the phenomenon under study. Ideally, an external group monitors the phenomenon, while data is extracted by those executing the activities.
	Action Research	Empirical method that aims to solve real world problems, while simultaneously studying the experience to solve a given problem.
Artificial Setting	Laboratory Experiment	(in ESE Wiki, Laboratory Study) Laboratory Experiments are characterized by taking place in acontrolled environment created for the purpose of research. Thus laboratory experimentsdo not necessarily have to take place in dedicated “laboratories” as such but canbe conducted in various controlled environments. If the experiment from the point of view of controlhas few subjects, it must be categorized as Case Study [Wohlin, M. Höst, P. Runeson, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering: an introduction*. Norwell, Massachusetts: KluwerAcademicPublishers, 2000]
Environment independent setting	Survey research	A comprehensive research method for collecting information to describe, compare or explain knowledge, attitudes and behavior. A survey is often investigations performed in retrospect, when, for example, a tool or technique, has been in use for a while.
	Applied Research	Builds on trial and error on the basis of the researcherscapabilities of reasoning through intuition, experience, deduction and induction. Typically the desired goal or outcome of the research process is known in termsof requirements on some level of abstraction, but methods or techniques foraccomplishingthis outcome are unknown and thus sought through applying potentially relevantresearch.
	Basic Research	Doing basic research, researchers develop new theories or study well-known problemsto which neither specific solutions nor methods for accomplishing solutions are known
	Normative Writing	This category is used to include non-research writings about a phenomenon. It covers aspects like concept development, ideas, and suggestions.

9 Annex III - Software Systems category

In section 2.6 the "Software Category" extraction data must be filled in with one of these values. This classification for Software Systems Category is taken from: R. Pressman, *Software engineering : a practitioner's approach*. New York: McGraw-Hill Higher Education, 2010.

System software— A collection of programs written to serve other programs.Some system software (e.g., compilers, editors, and file management utilities) processes complex, but determinate, information structures. Other systems applications (e.g., operating system components, drivers, networking software, telecommunications processors) process largely indeterminate data. In either case, the systems software area is characterized by heavy interaction with computer hardware; heavy usage by multiple users; concurrent operation that requires scheduling, resource sharing, and sophisticated process management; complex data structures; and multiple external interfaces.

Application software— Stand-alone programs that solve a specific business need. Applications in this area process business or technical data in a way that facilitates business operations or management/technical decision making. In addition to conventional data processing applications, application software is used to control business functions in real time (e.g., point-of-sale transaction processing, real-time manufacturing process control).

Engineering/scientific software—It has been characterized by "number crunching" algorithms. Applications range from astronomy to volcanology, from automotive stress analysis to space shuttle orbital dynamics, and from molecular biology to automated manufacturing. However, modern applications within the engineering/scientific area are moving away fromconventional numerical algorithms. Computer-aided design, system simulation, and other interactive applications have begun to take on real-time and even system software characteristics.

Embedded software— Resides within a product or system and is used to implement and control features and functions for the end user and for the system itself. Embedded software can perform limited and esoteric functions (e.g., key pad control for a microwave oven) or provide significant function and control capability (e.g., digital functions in an automobile such as fuel control, dashboard displays, and braking systems).

Product-line software—Designed to provide a specific capability for use by many different customers. Product-line software can focus on a limited and esoteric marketplace (e.g., inventory control products) or address mass consumer markets (e.g., word processing, spreadsheets, computer graphics, multimedia, entertainment, database management, and personal and business financial applications).

Web applications— Called "WebApps", this network-centric software category spans a wide array of applications. In their simplest form, WebApps can be little more than a set of linked hypertext files that present information using text and limited graphics. However, as Web 2.0 emerges, WebApps are evolving into sophisticated computing environments that not only provide stand-alone features, computing functions, and content to the end user, but also are integrated with corporate databases and business applications.

Artificial intelligence software— Makes use of non-numerical algorithms to solve complex problems that are not amenable to computation or straightforward analysis. Applications within this area include robotics, expert systems,pattern recognition (image and voice), artificial neural networks, theorem proving, and game playing.