Draft version 2 Revision history:
Giancarlo: - 07.02.03
Mike
Bernhard
Michele
Jose
Manlio
Leon
Giancarlo [Delete revision history]

The NAME Consortium

Proposal of the Structure of an Experience Framework for Agile Methodologies

José H. Canós[1], Mike Holcombe[2], Andrea Janes[3], Michele Marchesi[4], Leon Moonen[5], Manlio Reisoli[6], Bernhard Rumpe[7], Barbara Russo[3], Nicola Serra[4], Alberto Sillitti[8], Giancarlo Succi[3][1]

[1] Departament de Sistemes Informàtics i Computació, Universitat Politècnica de València

[2] Department of Computer Science, University of Sheffield

[3] Center for Applied Software Engineering, Free University of Bolzano-Bozen

[4] DIEE – Università di Cagliari

[5] CWI

[6] Datasiel S.p.A.

[7] Institut für Informatik, Technische Universität München

[1] Address for correspondence: Giancarlo Succi, Center for Applied Software Engineering, Free University of Bolzano-Bozen, Dominikanerplatz 3 Piazza Domenicani, I-39100 Bolzano-Bozen, Italy. E-mail: Giancarlo.Succi@unibz.it; tel: +39(0471)315-640; fax: +39(0471)315-649

[8] DIST – Università di Genova

Introduction

This document defines guidelines to experiment the effect of the introduction of (practices of) Agile Methodologies on the software development process.

The goals of these guidelines are three:

- Experiments should be replicable: replication is fundamental for the consolidation and validation of best practices in every applied discipline. Replication needs a standardization of the experimental process.
- The consolidated results should be promptly generalized and readily accessible.
- The whole Experience Framework should be "agile:" it should allow repository and standards updating, experiment replications and easy access.

In this document, we do not describe the universe of agile methodologies. Interested reader s should refer to existing literature and websites, partly listed in the NAME website http://name.case.unibz.it/.

Moreover, agile methodologies are often loosely applied, selecting few core practices. For this reason, we focus on the definition of the experiment rather than specifying the name of a methodology, in term of practices often used in Agile Methodologies.

This document contains the overall experience framework, structured as follows:

Structure of the Experience Framework

1. **Standards for data collection**
2. **Standards for data analysis**
3. **Standards for the set up of experiments**
   - 3.1. Experiment context and profile
   - 3.2. Goal Question Metric (GQM) of the experiment
   - 3.3. Experiment design
   - 3.4. Standards for the experiment data collection
4. **Standards for the actuation of experiments**
   - 4.1. Running of the experiment according to the specifications
   - 4.2. Storage of the data

4.3. Evaluation of the data

4.4. Storage of the results

Appendices are added dealing with specific issues related to this framework.

Appendix A contains the practices in the methodologies we considered

Appendix B is the template to define experiments

Appendix C contains acronyms and symbols used in the Experience Framework

1 4 Standards for data collection

We need to finalize this section.

A good analysis starts from valid data. Data validation and collection is a continuous process following all the phases of software development. Data need to be validated in the field, for example, interviewing experts of the project. On the other hand, one can increase accuracy of data validation providing the experts of guidelines for data collection.

Following the approach of (Basili *et al.,* 1984), we organize the data collection process along to six recommendations.

1. The data must contain information permitting identification of types of errors and change made
2. The data must include the cost of making changes
3. Data to be collected must be defined as a result of clear specification of the goals of the study
4. Data should include studies of projects from production environments, involving teams of programmers
5. Data analysis should be historical; data must be collected and validated concurrently with development
6. Data classification schemes to be used must be carefully specified for the sake of repeatability of the study in the sane and in different environment

A methodology in collecting data prevents subsequent misunderstanding and deviated results. Standards guarantee replications in data collection both in the same and in other experiments, reduction of effort of collection and traceability of data all along the development process.

Hereby we outline a guideline for gathering valid data following the previous recommendations. The first two aspects that should be taken into account are considering the material that is already available and doing a formative action on the environment.

Look at sources of information already available. A software company has several form of control of the software development: the accounting department (for example if the Activity Based Cost is used), an existing defects database, a customer service center or a configuration management system. (G. Succi, 2002) 'A lightweight Evaluation of a Lightweight Process' in (Marchesi et al., 2002).

For example, a company adopting XP may have user's story cards collection which may be source of information and may help in validating data.

Do a formative action on the environment. Data may be stored and forgotten or simply not considered relevant to the project. A formative action on members should encourage identifying correct data and using appropriate tools for collecting and validating them.

The guidelines should include

- Definition of data types and categories. It helps to distinguish and identify data in their classification.
- Techniques to extract data. Several approaches can be adopted and integrated. Agile practices recommend the use of automatic tools. One can find automatic tool useful for any phase of the software life cycle:
    - To trace employee' activity such as Hackystat (Johnson, 2001), Prom (CASE)
    - To trace and relate user's story cards
    - To trace modification requests such as Starteam
    - To test tools – JUnit, CUnit ….
- Standards for subjective data collection. Questionnaires and direct interviews are very often used in collecting data – for example on practices satisfaction – but can also be integrated in other methods in order to validate both methods and data.
- Standards for agile database. In gathering data one should always have in mind the kind of database he/she is going to build. A database collecting data from agile processes should reflect characteristics of flexibility, accessibility and data replication, see (ECWise, 2001).

From this guideline one should be able to delineate three or four types of data collection.

3 0 Standards for data analysis (TBD)

We need to finalize this section.

In this section standards must be set so that the data analysis is:

a. Replicable, ensuring robustness
b. Comparable, enabling generalizations of findings
c. Coherent, in line with the existing research

The findings are to be expressed in a clear and well-defined language.

We need

- To use a GQM
- To state the research hypothesis
- To look for patterns
- To set statistical methods to make inferences – Chi-square test, Kolmogorov-Smirnov test, Wilcoxon Mann Whitney test, Likelihood Ratio test
- To define template for findings collection
- To use automated tool for data analysis – statistics tools, algorithms for mining and identifying patterns of set of data (ECWise, 2001)

5 0 Standards for the set up of experiments

5.1 Experiment context and profile

The experiment needs to be modeled on the context. First we need to identify the context profile. It this way we can outline a replicable scenario and define the suitable competencies and roles to conduct the experiment.

5.1.1 Formative Research

A pre-test on the experiment environment detects the rate of difference caused by the treatment. A pre-test turns to be essential when it is hard to repeat the experiment or the amount of data is limited. A formative research action on the environment may help in defining the experiment standards and detect the pre-test scenario. On the other side it may cause a bias and produce consequent rival variables: people may become aware of the experiment goal.

The formative research would consider to major aspects:

- Project staff conducts a formative period of research including interviews to determine the knowledge present in the environment

- Using a standard protocol that is adapted to the local needs of the environment, project staff may also conduct qualitative, semi structured interviews with stakeholders in the environment.

Example

We report here two scenarios that can be successful to collect context information. They were discussed in [Goldman2002].

A successful software practice

- Customer comes with a basic problem
- The customer and the developer sit down together and define the scope of the scenarios to be written
- They work together to write between 3 and 10 short scenarios (usually with between 4 and 12 steps in each scenario)
- They review the scenarios with the project stakeholders
- The scenarios are used to drive further analysis or design (CRC cards, UML class diagrams, and so on)

...with automated info collection

- Developers create design model with scenarios
- A tool parses the model and generates simple test scenarios
- Developers or testers add more test-level detail to the test scenarios, to guide the creation of actual test code
- Later, the design model changes, and the same tool is run again to update the list of tests

The formative research protocol would help in retrieving the following three sections.

5.1.2 Background information about the industrial circumstances in which an empirical study takes place

The experiment context would involve software SMEs either competent or only interested in the Agile Methodologies.

In this section the research should specify

- Software company competence
- Software company standards – company standards process, quality assurance procedure, configuration management process
- Staff expertise and skills – with languages, agile methods, tools and application domain
- Technical support and facilities– platforms, components etc.
- Automatic tools for data collection or analysis – metric tools, log files etc.
- Company standards for reporting research context:

- Specification of taxonomy or ontology of context – e.g. XP vocabulary: test first, customer on-site, coach, stand up meeting
- Knowledge management – knowledge documentation template, knowledge repository, automatic tool, post-mortem analysis, evangelist, interviews, questionnaire, XP customer's manual

5.1.3 Background information about the experiment

- Kind of projects - pilot project, exploratory experiment, comparative project, project converted to XP
- Project applicative domain
- Project staff composition – number of teams, number of team components, coaches, work environment and staff geographical distribution, turn over
- Agile Methodology considered – XP, SCRUM, Dynamic Systems Development Methodology, and Feature Driven Development, Crystal etc.
- Start up practice or sequential order of practices – process length and iterations decomposition, development iteration length
- Product type
- Possible confounding factors for the specific empirical study – non-complete adoption of all the practices, customer different approach, project early suspension, pressure to ship, turn over

5.1.4 Information about related research

To collocate the experiment in the appropriate research context one needs to consider the state of art of the related research. The entire research picture could help in defining data collection and storing standards.

5.2 Goal Question Metric (GQM) of the experiment

Planning the experiment reduces costs and increase correctness of the results. Research hypothesis can be stated using a Goal Question Metric paradigm (Basili *et al.*, 1988, Wholin *et al.*, 1999). This procedure traces Goal, Question and Metric before starting the experiment and the data collection.

Goal	**Question**	**Metric**	**Feedback**

Analyze **the object of study** in order to **purpose** with respect to **focus** from the **point of view** in the following **context**.	List the questions in order to achieve the Goal.	To measure data or quality of the analysis, set a set of metrics according to data features and questions	What has been learned here? In order to clarifying the research for future use of it, sum up results in a "learned lesson" section.

5.2.1 Setting goals

To specify how goals are to be set, here below there is the template from (Basili *et al*., 1999)

Action {characterize, evaluate, predict, motivate, etc.}

Object of study {process, product, model, metric, etc.}
Purpose in order to {understand, assess, manage, engineer, improve, etc.} it
Perspective examine the {cost, effectiveness, correctness, defects, changes, product measures, etc.}
Point of view {customer, developers, managers}
Context specify the environmental factors, including process factors, people factors, problem factors, methods, tools, constraints, etc.

5.2.2 Setting Metrics

Metrics are chosen to measure the object of study and answer to questions. Three aspect may have been considered:

- Consider the most used metrics
- Identify appropriate metrics
- Consider metric integration

5.2.3 A comprehensive example

The example here below specifies defines the metrics for an experiment on pair programming. It is taken from (Succi *et al., 2002).*

Goal	Question	Metric	Feedback
Analyze **pair programming** in order to **evaluating it** with respect to **job satisfaction** from the point of view **software developers** in the **development of software systems**	Did the developers using pair programming experience higher job satisfaction than those not using pair programming?	Rate of job satisfaction.	It appears that pair programming effects job satisfaction of developers.

5.3 Experiment design

By the theory of Experimental Design (Campbell *et al.*, 1963) we can choose among several designs of an experiment according to the particular context in which the experiment is conducted. Very often experimental design cannot be adopted and quasi experimental design is more feasible. The choice may be connected with the amount of available data: quasi- experimental design is suitable for a limited amount of data or when a randomization process on the data is not feasible. Interested reader should refer to the mentioned book.

5.3.1 Theoretical Foundation of the Intervention

In this section the experiment scenario, treatment and metrics are set in order to determine the appropriate (quasi-)experimental design.

- Define the experimental unit
- Identifying the population being studied
- Declaring the methods used to reduce bias and determine sample size
- Stating the rationale and technique for sampling from that population
- Defining the process for allocating and administering the treatments (Campbell *et al*. 1963)
- Identifying the possible rival variables to the interpretation of the experiment, which may cause bias
- Rejecting inadequate hypothesis
- Identify measures in terms of their relevance to the objectives of the empirical study

5.3.2 Distribution of Materials by Community Network Members

The research is conducted within a research network. Distribution of documents to the community would help reducing bias and possible rival variables. Common standards are defined.

5.4 Standards for the experiment data collection

Data collection is a rather delicate matter and a particular care to follow the standards should be taken.

5.4.1 Data collection instrument

- Automatic tools for data collection peculiar to the experiment

5.4.2 Data collection schedule

Data collection is a continuous process concurrent to the development process. We need to identify phases of data collection according with software life cycle. Each phase would give different categories of data.

5.4.3 Data collection Start Up

- Trial period to test forms
- Kick-off session
- Checking forms for correctness and completeness
- Storing form data

5.4.4 Building a measurement support system

- Spreadsheets
- Statistical tools
- Database applications
- Presentation tools

5.4.5 Aggregate the data in analysis sheets

- Raw data
- Processed data
- Graphs and tables

7 4 Standards for the data storage (TBD)

In our process we need to provide recommendations on identifying database profiles for data and for results. In line with the agile methodologies agile databases should be preferred. Standards must reflect features of the agile universe. Literature is scarce for agile databases for data (see ECWise, 2001), and it is null for agile database for results.

In fact managing information is a quite hard matter since it is difficult to quantize knowledge.

7.5 Storage of the data

Storing the data we need to take into account three characteristics of the repository we are going to build.
It should allow

- Continuous integration
- Collective ownership
- Collaborate with developers

7.6 Storage of the results

As before we need to consider two features of the repository we are building.
I should allow

- Continuous integration
- Collective ownership
- Outsourcing
- Accessible for reuse

References

[Basili1988] V.R Basili., D. Rombach "The TAME Project: Towards Improvement-Oriented Software Environments," *IEEE Transactions on Software Engineering*, 14, No. 6, June, pp. 758-773, 1988.

[Basili1984] V.R. Basili, D.M. Weiss "A methodology for collecting valid software engineering data," *IEEE Transactions on Software Engineering,* 10, No. 6, pp. 728-738, 1984.

[ECWise2001] 'Toward an Agile Database' Whitepaper, 2001.

[Kitchenham2001] B. Kitchenham, S. Pfleeger, L. Pickard, P. Jones, D. Hoaglin, K. El-Emam, J. Rosenberg, "*Preliminary guidelines for Empirical Research in Software Engineering*", National Research Council for Canada, 2001.

[Goldman2002] R. Goldman "Scenarios," presented in *OOPSLA2002 workshop on Distributed eXtreme Programming, Tackling the Discovery Costs of Evolving Software Systems and Pair Programming Explored*, Seattle, MA, 2002.

[Report2002] "Community-Level Prevention of Human Immunodeficiency Virus Infection Among High-Risk Populations: The AIDS Community Demonstration Projects," *Journal of the American Medical Association* MMWR Recommendations and Reports, 45, No. RR-6, 2002.

[Succi2002] G. Succi "A lightweight Evaluation of a Lightweight Process," in M. Marchesi, G. Succi, D. Wells, L. Williams *Extreme Programming Perspectives*, Addison Wesley Professional, 2002.

[Succi2002] G. Succi, M. Marchesi, W. Pedrycz, L. Williams "Preliminary Analysis of the Effects of Pair Programming on Job Satisfaction," *Proceedings of the 3rd International Conference on XP and Flexible Processes in Software Engineering*, Alghero, Italy, 2002.

[Wohlin 1999] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, A. Wesslén *Experimentation in Software Engineering: An Introduction,* Kluwer Academic Publishers, December, 1999.

Appendixes

A. List of Practices in Agile Methodologies

B. MS Word template to help the classification and the description of experiments

C. Acronyms and symbols used in the Experience Framework

Appendix A: List of Practices in Agile Methodologies

<Just started>

Methodologies	**Practices**
eXtreme Programming	Planning Game
eXtreme Programming	Small Releases
eXtreme Programming	Metaphor
eXtreme Programming	Simple Design
eXtreme Programming	Testing
eXtreme Programming	Continuous Integration
eXtreme Programming	Pair Programming
eXtreme Programming	Collective Ownership
eXtreme Programming	Refactoring
eXtreme Programming	40-Hour Week
eXtreme Programming	On-Site Customer
eXtreme Programming	Coding Standards
Todo: SCRUM	
Todo: Agile Modeling	
Todo: Crystal	
Todo: Feature Driver Development	
Todo: Dynamic Systems Development Method	
Todo: Adaptive Software Development	

Appendix B: MS Word template to help the classification and the description of experiments

<To do>

Appendix C: Acronyms and Symbols used in the Experience Framework

<To do>