AFFECTIVE AGILE DESIGN A PROPOSAL FOR A NEW SOFTWARE DEVELOPMENT MODEL

[1]ALESSANDRA PIERONI, [2]NOEMI SCARPATO and [3]MATTEO SCORZA

[1]Guglielmo Marconi University, Dept. of Innovation and Information Engineering, Rome, Italy
[2]San Raffaele Roma Open University, Dept. of Human sciences and Promotion of the quality of life, Rome, Italy [3]Fastweb SPA, Milan, Italy

E-mail: [1]a.pieroni@unimarconi.it, [2]noemi.scarpato@unisanraffaele.gov.it, [3]maffeo74@gmail.com

ABSTRACT

This paper deals with two apparently distant and well-known technologies: the *agile* development methodology and the *affective design* discipline. The aim of the paper consists of investigating how to combine the two technologies in order to obtain a new software development model: the *affective agile design*. The innovative software development model, indeed, leverages agile - Scrum Framework guidelines and recalls affective design practices that consist of collecting and analyzing emotional feedback during the running process. The proposed model, furthermore, address firmly people-oriented principles, that have opened up a new front in regards of the needs of the productive path, calling in a loud voice the initiation of affective design practices. The main pillars of the proposed *agile affective design* model are the following: defining the points of the process where iteration with individuals is strong, intercepting their emotional expressions, analyzing their useful contributions and proceeding with systematic affective analysis leading to an optimized process of change management.

In order to measure the validity of the proposed model, a real instance of a software developed by using the basic agile methodology at Fastweb Spa has been studied. The results have demonstrated that the benefits obtained by the use of the proposed innovative software development model overcome the main *pain-points* identified in the use case.

Keywords: *Agile Methodology, Affective Design, Affective Computing, Scrum, Software Development Models, Artificial Intelligence.*

1. INTRODUCTION

Agile development methodology and Affective Design discipline are decidedly people-oriented development fields that have brought the human figure to the fore, highlighting the importance of the individual, of his emotions, of his emotional states, of his interactions with other human beings and with the surrounding environment: an emotional wave that cannot be ignored, but ridden instead, taking full advantage of its benefits.

The idea behind this article is finding common points between these two worlds, Agile and Affective Design, defining a new software development model able to take advantages of the path indicated by the former and by the correct direction traced by the latter.

This innovative proposed development model, called *affective agile design*, is based on the agile principles of Scrum Framework: following the steps designed by the process it activates affective design actions within its life cycle with the goal of intercepting and managing the emotional contributions expressed by all individuals involved in the process.

In the *affective agile design* model, three points called Affective Points are identified within the Agile development process to activate Affective Design practices: the result of the Affective Design application leads to the activation of improving actions that have a direct impact on the ongoing Agile development process.

In order to measure the benefits of the proposed innovative development model, a test to a real Agile Software Development Case, created at

Fastweb spa, has been performed, as described in the following sections.

2. AGILE METHODOLOGY

In February 2001 at a ski resort in the State of Utah, 17 independently-minded software practitioners from the world of information technology met to discuss the future of software development in a stimulating and relaxing environment. Thus, Agile Manifesto was born, with its values and its principles [1] decidedly people-oriented, which exalt the human figure with his interactions within and outside the development Team:

- Customer Collaboration over Contract Negotiation;
- Individuals and their interactions rather than processes and tools.
- Build projects around motivated individuals. Give them the environment and support their needs, and trust them to get the job done.

The development Team is seen as a single coordinated entity and formed by people who must work together pushing step by step in the same direction to reach the shared goal, the creation of the project.

Agile methodology leaves behind totally predictive reality with rigid planning of each development phase, downstream of a complete definition of all specifications and it puts itself forward as an adaptive model with values and principles capable of following requirements evolution, focused on the ability to quickly adapt to change in the surrounding reality.

The Agile methodology is a breaking point with past tradition (see fig. 1), strong in the concepts of Creating Value [2] that it is translated in the iterative and incremental development by delivering working software in a short time with all the features chosen directly by the customer and of Change Management [3] that it is no longer seen as an obstacle but as an opportunity to build and sharpen the project for real business expectations, creating preconditions for a more compatible solution.

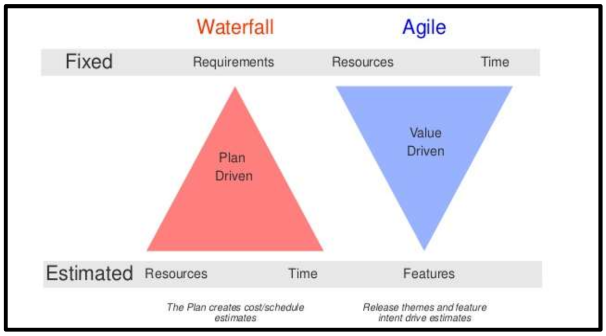

Fig. 1: Comparison between Waterfall and Agile models

Software is developed by human beings and must work for humans. The human figure is central throughout the development process.

Through the study of historical evolution of software engineering methodologies, it has been highlighted how the valorization of human factors in the development process is directly proportional to the effectiveness of the methodology in the real world [4].

People orientation in Agile Methodology also results in a radical change in the traditional requirement engineering process by evolving the definition of the requirement itself that becomes a dialogue tool between the Customer and Development Team.

Language used, by technicians and little understood by the customer and users of the product, becomes natural, clear to all the participants: the traditional requirement thus becomes the User Story (US), the story told by the User, as shown in figure 2. The structure of the User Story is simple and does not contain all the details useful to the implementation of the required functionality but it serves rather to initiate a continuous and collaborative dialogue between the Development Team and business:

- As I [rule];
- I Want [something];
- So that [benefit].

Agile Methodology does not impose any rules for drafting User Stories, but it suggests guidelines (I.N.V.E.S.T) to be effective and coherent with the new development process:

- *Independent*: independent of each other so that they can be developed in any order;
- *Negotiable*: do not contain too much detail, a right agreement between customer, developer and tester;
- *Valuable*: create value for its end user;

- *Estimable*: always be estimable;
- *Small*: be small enough to fit into a Sprint;
- *Testable*: ensuring what is developing is verifiable and testable.

Fig. 2 – User Stories

Furthermore, the Agile methodology can be considered as a collection of all development techniques that revolutionize traditional methods and share values and principles of Agile Manifesto.

In this sense, Scrum Framework is considered one of the main and most widely used Agile methodologies, which predicts, in compliance with an iterative and incremental approach, that the entire Project Stream will be divided into fast-working Blocks, called Sprint [5].

At the end of every Sprint, an incremental software version which is fully working, is delivered to the customers, until the final release composed by all the required features.

In figure 3, the development process, that will perform part of the innovative model, has been highlighted:

- Sprint planning – the Team is committed to developing User Stories belonging to the Sprint Backlog;
- Sprint review – presentation of incremental software Sprint release;
- Stand up meeting - moment of sharing for internal Team members ("What I did yesterday, what do I do today, what obstacles I found");
- Retrospective - sharing knowledge of introspection and adaptation.

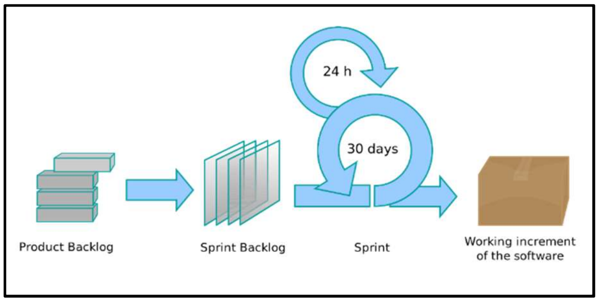

Fig. 3 - Sprint

Many are the scenarios in which the Agile approach could support and ease also the development of complex software, as for example, software for distributed simulation modeling [26-31], that are inherently complex systems. The collaboration between the agile and the affective design, may envisage a new generation of designing this software.

3. TOWARDS AFFECTIVE DESIGN

In September 1997 Rosalind W. Picard, an eclectic electric and computer engineer published a book entitled "Affective Computing" [6].

The book is a watershed in the world of Artificial Intelligence research, tense until then in trying to create intelligent machines capable of imitating human behavior by omitting the emotional sphere at the expense of a rational approach.

Many attempts are been provided to realize artificial intelligent technologies [13][14], and these technologies should been applied in very different field, such as legal domain [15][16], earth observation [17], Cultural Heritage [18], Medicine [19][20][21], IoT [22] and Internet of Vehicles [23][24][25].

Artificial Intelligence was focused on the most rational aspect, ignoring or entrusting a marginal role to emotions in developing artificial responses to "thinking machines." Richard Cytowic's neurological essay "The Man Who Tried the Shapes" doubts that the perception process is so rational and direct as it imagined in those years; furthermore, the themes of neuroscience about subjects suffering from "frontal syndrome", incapable of experiencing emotions in performing daily actions, clearly indicate how reasoning and emotion are actually two inseparable worlds.

Affective computing takes into account how emotions play a key role in decision-making in all learning-related activities and affect very rational mechanisms such as cognition, perception, and communication. Therefore, Affective computing aims to create intelligent machines capable of recognizing and exhilarating emotions, able of interacting with users and the surrounding environment by analyzing and interpreting emotional inputs instead of ignoring them.

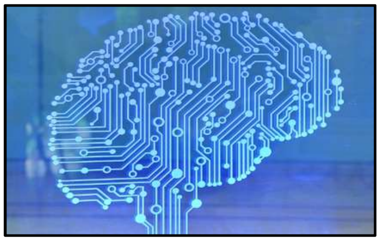

Fig. 4 – Artificial Intelligence

"We do not have to wonder if intelligent machines can have emotions, but if those machines can be intelligent without them" [7].

Affective Computing thus becomes an interdisciplinary field of research across computer sciences, psychology, neurology and cognitive sciences in an ongoing attempt to assign computers human capabilities, from interpretation to observation, to generation of influencing abilities, with the goal of increasing the quality of human-computer communication. Study of emotional input analysis [8] leads to the ability to represent emotions through two parameters traced in the Emotional Cartesian Space (ECS):

- Valence - represents the level of pleasantness and can range from Negative to Positive;
- Arousal - represents the level of excitation and can range from On to Off.

Starting from emotions, affective inputs, the ECS is used to trace emotions and to find their relative-moods: excited, tense, annoyed, happy, serene, sad, calm, strained, relaxed.

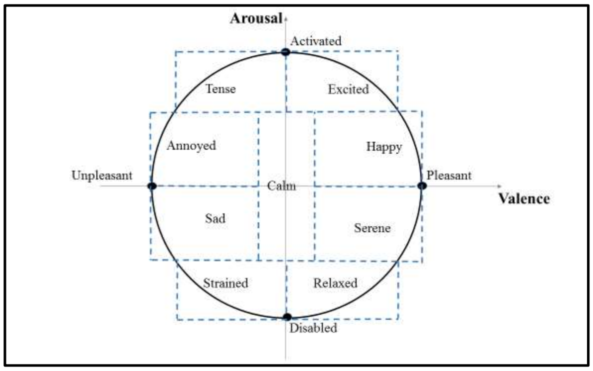

Fig. 5 - Emotional Cartesian Space (ECS)

Affective Computing research studies lead to the definition of a more specific field called Affective Design, which has as its main scope the user-interface design that allows man-computer interactions where the machine could process emotional information and respond to the surrounding environment [9].

The user is able to communicate his emotions and moods in a natural way; the machine is able to transpose, process and manage information received through the interface and replicate with activities aimed at improving the interaction that has just occurred [10].

Unimodal or multimodal [11] affective inputs can be of a different nature:

- visual - through facial expressions;
- audio - through spontaneous and natural dialogues, voice recordings;
- textual - through written contributions.

In this contest, it is possible to image an Affective Network. Systems and objects that independently exchange information gathered to build a shared network of detection and response. Detect emotional data and share information across all involved systems to provide a unique response and anticipate user's needs at all levels of experience [12].

This is how Affective Design predominantly comes into the development process of a software product, detecting Users' emotional responses during the design phase, addressing developments based on the detected emotional feedback and timely correcting features before being put into production.

We highlight three phases that will be useful for our model:

1. *Monitoring phase* - Actively intercepts emotions by scanning the surrounding environment or passively receiving a set of unimodal or multimodal inputs; chooses

which signals to analyze and sends them to the analysis phase;

2. *Analysis phase* - Analyzes data recorded in the first monitoring phase, identifying emotions expressed by individuals involved, calculating their relative moods and their personality;
3. *Processing phase* - Works out a responsive response action suitable for the context.

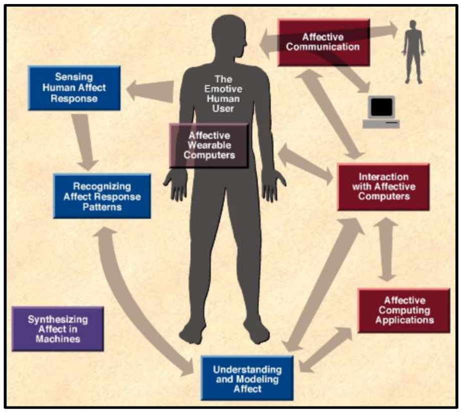
Affective Communication
Sensing Human Affect Response
The Emotive Human User
Affective Wearable Computers
Recognizing Affect Response Patterns
Interaction with Affective Computers
Affective Computing Applications
Synthesizing Affect in Machines
Understanding and Modeling Affect

Fig. 6 – Research areas of Affective Computing

4. AFFECTIVE AGILE DESIGN

As said above, the Agile methodology and the Affective Design discipline are firmly people-oriented research fields that have brought the human figure to the forefront, highlighting the importance of the individual, his emotions, his affective states, his interactions with other humans and the surrounding environment.

Therefore, it is natural and necessary to find common points between the two worlds within an affective perimeter that emphasizes the figure of the individual, his emotions in proposing a new software development model that exploits the path indicated by the former (Agile) and finds the correct direction through the application of the latter (Affective Design).

Based on Agile principles of Scrum Framework a new software development model is defined, which activates within its life-cycle Affective Design actions where it is necessary to detect and analyze emotional feedback of all the actors involved and to initiate corrective action of the running process.

The Inspect & Adapt Agile concept is *resumed* and *extended* to become the guideline of this new model of Affective Agile Design.

Looking around, listening emotionally to the inputs from the surrounding environment, finding useful and necessary information on the development path, understanding what really works and what is creating an emotional impediment, taking on expectations and proposing concrete and direct action to change negative state.

Agile methodology needs humans, individuals, emotions. All the decisions taken during the life cycle of development process are the result of affective reasoning, a set of moods that lead to one road rather than another, driving the group towards the end goal, from the choice of priorities, to distribution of workloads, from acceptance of User Stories to the good result of their work.

The unbalanced approach to firmly people-oriented principles opens up new challenges within the productive path, calling in a loud voice the initiation of Affective Design practices: defining the points of the process in which iteration with individuals is strong, intercepting their emotional expressions, analyzing useful contributions and proceeding with systematic affective analysis that leads to a plan for processing and managing the change.

The new Affective Agile Design model tries to cover those new requirements, as seen in figure 7.

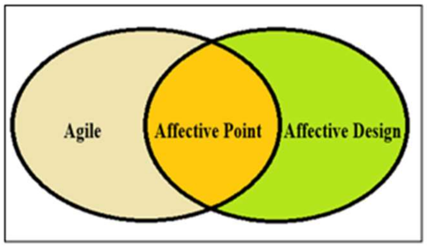

Fig. 7 – Affective Agile Design

Firstly, it has been necessary to define the so-called Affective Points as the points of contact between Agile and Affective Design domains: they are the points where Affective Design practices are launched within the life cycle of the Agile development process.

Emotional inputs by individuals involved are traced, collected, analyzed and elaborated, and the result of Affective Design's application leads to the activation of improvement actions that have a direct impact on the ongoing development process.

In Affective Agile Design model, we have identified 3 affective points:

1. Post-It Lot - triggered during each Sprint Review - Processed results are discussed during Sprint Planning.

Through Post-It Lot, Affective Agile Design model is able to trace emotional information from every Sprint Review participant, to analyze these emotions, to test results of the analysis by comparing them with related KPI and possibly to trigger corrective action.
Post-it Lot name derives from “Parking Lot”, a definition used in Agile world to indicate a board over which questions from participants during a meeting are habitually written. All participants are asked to leave a post-it in each parking area for every User Story shown by the Team during the Sprint Review by answering the following questions:

- "Do you accept the User Story?"
- "Does the User Story meet your expectations?"

Post-It Lot board contributions are collected by the Scrum Master at the end of the Review and digitally scanned in order to start an analysis of Affective Design.

2. Affective Simulation triggered during each Sprint Review - Processed results are discussed during Sprint Planning.

Affective Agile Design model predicts that at the end of each Sprint periodic sessions of Business Simulation will be set up, where the new portion of software created and presented will be made available in a demo environment for End Users in order to test the key features.
Affective Simulation consists of tracing the emotional information of Business Simulation’s participants, analyzing their emotions, testing the analysis results by comparing them with the KPI to monitor and possibly trigger corrective actions for subsequent Sprint.
For each Sprint, User Stories with greater Business Value are selected and their developed features are presented through use cases in a demo environment where Users can experience it in a guided manner.
An interface developed to monitor User tests records their sessions by tracking specific parameters:

- Number of mouse clicks;
- Processing times;
- Number of reworking;

At the end of each use case, a free response question about the functionality just tested is offered to the User:

- "Did you like the test?"

User's textual comment is free and serves to record his emotions during work sessions to start an analysis of Affective Design.

3. Team Mood - triggered during Daily Meetings - Processed results are discussed during the Retrospective meeting.

It is important for the Team Leader to understand his colleagues’ affective state, as the developer's mood or its swing may have negative impacts on the process of running activities, producing significant slowdowns with heavy quality repercussions for the developed software.
Team Mood consists of tracing Team members’ emotional information during all steps of the development process, analyzing their emotions, testing results of the analysis by comparing them with related KPI and possibly triggering corrective actions.
Team members are asked to leave a post-it on Team Mood board with a face drawn smiling, neutral, or sad depending on the developer's mood, and a free text comment describing personal feelings about the meeting.
Team Sprint Productivity is also recorded:

$$= \frac{\text{Effective quantity User Stories developed}}{\text{Expected quantity User Stories developed}} \%$$

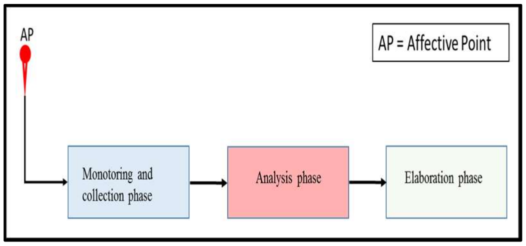

Fig. 8 – Affective Point phases

Each Affective Point has three phases, as seen in figure 8:

1. Monitoring and collection phase - Emotions expressed by individuals involved in the development process are identified and traced; Affective Agile Design model presents a different and specific way of collecting affective information for each of the three

Affective Points. A dedicated board for Post-It Lot, a graphic interface for Affective Simulation, a specific board for Team Mood.

2. Analysis phase - Emotions collected during monitoring and collection phase are analyzed and remapped into emotional states that lead to a specific personality belonging to a unique entity, representing all the individuals involved. Analysis phase is homogeneous for each of the three Affective Points and it consists of computing the Affective Scheme, divided into 3 main Task:

   - Emotions Calculation – Emotion (Valence, Arousal);
   - Moods Calculation - excited, tense, strained, happy, serene, sad, calm, annoyed, relaxed;
   - Personality Calculation: Promoter, Fence-Sitter, Demotivator, Detractor (see fig. 9).

3. Elaboration Phase - Improvement actions of the ongoing development process are addressed based on the personality identified in the analysis phase. This phase presents a specific and different processing mode of affective information for each of the three Affective Points.

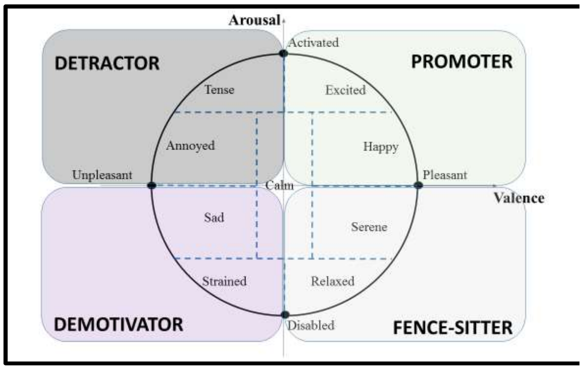

Fig. 9 – Personalities

5. USE CASE FASTWEB

In January 2016, the largest provider of broadband telecommunication services in Italy (Fastweb spa), initiates an internal process of digital transformation, starting with the definition of a corporate vision and mission that indicates a common goal and direction to be followed during the whole journey, dictating internal guidelines and highlighting the human factor.

Fastweb's transformation process emphasizes the most human aspect of the company, consolidating and sharing principles in which referring to every moment of the journey as a solid base for all the most technical aspects of the project.

The transformation covers all internal systems (Business Support Systems) and OSS (Operational Support Systems) and, in particular, a new Service Management system is implemented with the adoption of Agile methodology – Scrum Framework.

The project Stream starts with an Inception phase of 8 weeks, followed by an iterative cycle of 10 Sprints of 4 weeks each one; at the end of each Sprint an incremental Software Release is expected in a test environment.

Within the project flow, 3 software releases in the production environment are planned with the following subdivision (see fig. 10):

- Release 1 - at the end of Sprint 5;
- Release 2 - at the end of Sprint 8;
- Release 3 – at the end of Sprint 10;

After the development of every Sprint User Stories are presented during the Sprint Review to an audience of 60 people.

After each of the 3 releases in the production environment, 3 Business Simulation sessions are organized: the number of participants configured to access test systems in test environment is 50 users.

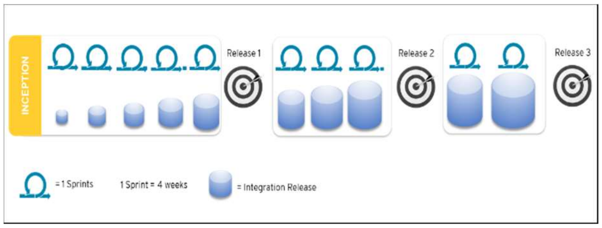

Fig. 10 – Project Stream planning

During Business Simulations, Users are able to test features that have been developed.

The number of participants configured to access systems in the test environment is 50 users.
Starting from a 128 User Stories - Product Backlog, the Project Stream has followed a very regular trend in terms of the number of User Stories created (placed in Done) in each Sprint: after a first Sprint of setting, every Sprint has stabilized with regular production around the average of 12.8 User Stories, as illustrated in figure 11.
The main benefits identified was the following:

- Communication between groups - The simple language and structure used to write User Stories made easier to have a dialog between the two macro areas of the company, one more connected to Business (Users and Stakeholders) and one more technical (Information Technology and Development Team);

- Definition of requirements - A long-term project (5 years) such as Fastweb's Digital Transformation made it difficult to consolidate all requirements before starting to develop and to produce and release all the functionalities of a new system of Service Management in a single release; production of features in iterative and incremental cycles made it easier to plan the Development Roadmap and increased the stakeholders buy-in and business representatives' involvement who have seen the product growing Sprint after Sprint;

- Change management - The structure in development Sprints has been able to handle and promptly direct changes of functional requirements and business priorities.
Example: It has been asked to bring forward the development of some features from Release 3 to Release 2 in order to handle a Public Ban;

- Team building - The Development Team worked closely with the entire Scrum Team and Stakeholders involved in the process, creating a strong relationship in the partnership that has led to sharing the same goals.

On the other hand, the main pain points identified was the following:

- Rework - Some User Stories during the Inception were not properly defined and lead to incorrect or incomplete development at the delivery stage: it is very difficult to intercept the error without having structured feedback and the Development Team was forced to perform reworking by reprocessing User Stories already developed (put in Done) and increasing effort into subsequent Sprints;

- Testing environment - Business Simulation sessions available to users were too limited in time and there was a lack of a system able to systematically track feedback from User Experience.

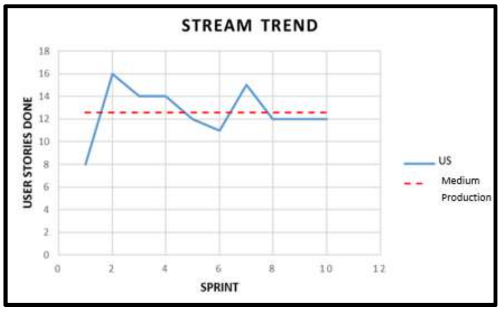

Fig. 11 – Stream trend

6. APPLICATION OF THE AGILE AFFECTIVE DESIGN MODEL

In this section, the application of the *agile affective design model* to the Fastweb Software Development Use Case will be described.
During the first phase, three Affective Points (see Fig. 12) have been identified.
The affective contact points where it has been possible activate the Affective Design practices, within the life cycle of each Agile Sprint, are the following:

- AP01 - Affective Point 01 (Post-IT Lot) during the Sprint Review;

- AP02 - Affective Point 02 (Affective Simulation) during Business Simulation session;

- AP03 - Affective Point 03 (Team Mood) during daily meetings.

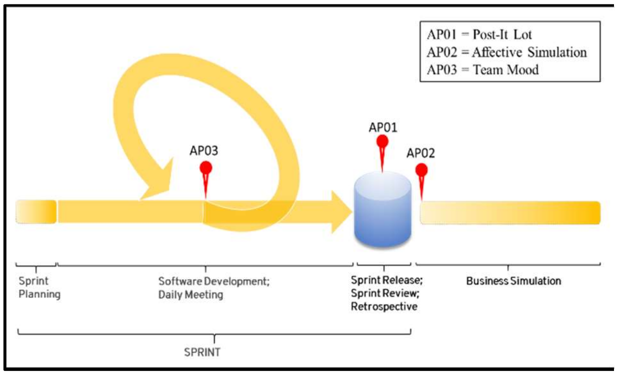
AP01 = Post-It Lot
AP02 = Affective Simulation
AP03 = Team Mood
AP03
AP01
AP02
Sprint Planning
Software Development; Daily Meeting
Sprint Release; Sprint Review; Retrospective
Business Simulation
SPRINT

Fig. 12 – Affective Point mapping

Concerning the AP01 (Post-It Lot), we have tried to apply Affective Agile Design model to the first Sprint of the Project Stream.

Analyzing more in details this Affective Point (Post-it Lot), it is possible to define the following three phases:

1. Collection and Monitoring phase;
2. Analysis phase;
3. Processing phase

Collection and Monitoring phase

The audience at the Sprint Review (about 60 people) would be invited by Scrum Master to leave a post-it contribution on the dedicated board for each User Story by answering the following questions:

- "Do you accept this User Story?"
- "Does the User Story Meet Your Expectations?"

At the end of the Sprint Review, the Scrum Master would be able to digitally record all the contributions posted on the Post-It Lot Board:

- 60 (people) x 8 (US) = 480 contributions;

Contributions would then be launched at the Analysis phase.

Analysis phase

Emotions Calculation - Valence Level. We assume the Fastweb population present at the first Sprint Review has expressed the following judgements (see Table 1) about the User Story E128.2 - "Customer Identification":

Table 1 - Valence level calculation

Quantity	Percentage%	Judgment	Valence
10	16,67%	Accepted	Pleasant
46	76,67%	Not accepted	Unpleasant
4	6,67%	I don't know	Neutral

The value taken into consideration is a level of Valence of 76.67% - Unpleasant, to trace the emotion over the Emotional Cartesian Space.

In order to calculate the Emotional contributions – Arousal Level - we suppose that the result of the emotional contributions of Fastweb people present at the first Sprint Review is a level of excitement tending to “disabled” values.

Mood Calculation. The Arousal and Valence levels are calculated form the coordinates of the emotional points in the Emotional Cartesian Space. As shown in figure 13, we have a single emotional value (x) expressed about the evaluated User Story, where the *Emotion* value is expressed as a function of Valence and Arousal.

The *Emotion* (x) calculated and traced in the ECS, would result within an area that represents a specific mood, in our case the mood of "Sad", as illustrated in fig. 13.

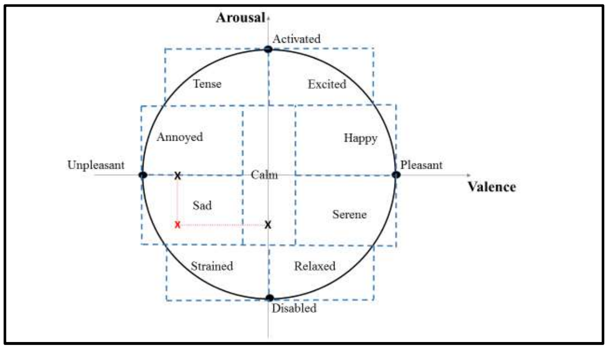

Fig. 13 – Mood computing

At this point is possible perform the Personality Calculation. Indeed, the mood found in the preceding phases leads to associating a "de-motivator" personality to the group representing Fastweb's audience present at the first Sprint Review.

The model would identify the improvement actions chosen based on the personality detected for this scenario: de-motivator.

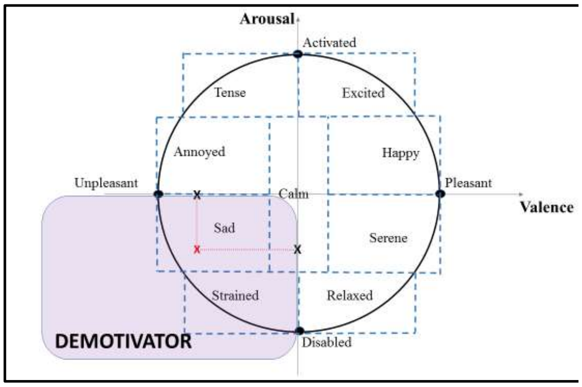

Fig. 14 – Personality computing

Processing phase
In this last phase, improving actions have to been performed. The solution chosen to develop User Story features is not clear, the user does not perceive a processing simplification. Thus, it is necessary to start rework of the User Story in subsequent Sprints: an immediate re-planning is required to initiate corrective actions in order to improve the Graphic User Interface and improve the User Experience making it simpler and more immediate.

7. CONCLUSIONS

In this paper, we have illustrated two different and well-known technologies: the *agile* development methodology and the *affective design* discipline. Has been demonstrated how the above-mentioned technologies may be combined in order to obtain an innovative software development model: the *affective agile design*.

To support the validity of the proposed model, the application to a real case study, has been illustrated. In particular, has been proved that the proposed model is able to leverage agile guidelines while recalling affective design practices, collecting and analyzing emotional feedback during the running process.

The following main steps has been illustrated and applied to the real software case: defining the points of the process where iteration with individuals is strong, intercepting their emotional expressions, analyzing their useful contributions and proceeding with systematic affective analysis leading to a process of change management, effective and immediate by intercepting in good time activities of rework to be planned within the project. The obtained results have demonstrated that the benefits provided by the use of the proposed innovative software development model overcome the main pain-points identified in the use case.

REFERENCES

[1] THE BLOKEHEAD, Scrum - Guida Pratica Definitiva alle Pratiche della Metodologia di Sviluppo Agile!, Babelcube Inc., 2016.

[2] R. MIRONOV, The Agile Product Manager - Product Owner Dilemma, Mironov Consulting, 2012.

[3] K. BECK, Extreme Programming Explained: Embrace Change, Boston, Addison-Wesley, 2004.

[4] J. LIN, Human factors in Agile software development, 2015.

[5] K. WATERS, All about Agile, Paperback, 2012.

[6] ROSALIND W. PICARD, Affective Computing, The Mit Press, 1999.

[7] M. MINSKY, La società della mente, Adelphi, 1985.

[8] DONGRUI WU, Optimal Arousal Identification and Classification for Affective Computing Using Physiological Signals: Virtual Reality Stroop Task, IEEE Computer Society, 2010.

[9] C. REYNOLDS and R.W. PICARD, Designing for Affective Interactions, MIT Media Laboratory

[10] J. JIAO and Y. ZHANG and M. HELANDER, A Kansei mining system for affective design, Elsevier

[11] S. PORIA and E. CAMBRIA and R. BAJPAJ and A. HUSSAIN, A review of affective computing: From unimodal analysis to multimodal fusion, Elsevier.

[12] M. PIERONI and L. RIZZELLO and N. ROSINI and D. MAZZEI, Affective Internet of Things: Mimicking human-like personality in designing smart-objects, Conference Paper.

[13] M. T. Pazienza, N. Scarpato, A. Stellato, and A. Turbati, “Din din! The (Semantic)

Turkey is served!," in CEUR Workshop Proceedings, 2008, vol. 426.

[14] M. T. Pazienza, N. Scarpato, A. Stellato, and A. Turbati, "Semantic Turkey: A Browser-Integrated Environment for Knowledge Acquisition and Management," Semant. Web, vol. 0, pp. 1–14, 2011.

[15] M. Bianchi, M. Draoli, G. Gambosi, M. T. Pazienza, N. Scarpato, and A. Stellato, "ICT tools for the discovery of semantic relations in legal documents," in CEUR Workshop Proceedings, 2009, vol. 582.

[16] M. T. Pazienza, N. Scarpato, and A. Stellato, "STIA*: Experience of semantic annotation in Jurisprudence domain," in Frontiers in Artificial Intelligence and Applications, 2009, vol. 205, pp. 156–161.

[17] F. Fallucchi, M. T. Pazienza, N. Scarpato, A. Stellato, L. Fusco, and V. Guidetti, "Semantic bookmarking and search in the earth observation domain," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2008, vol. 5179 LNAI, no. PART 3, pp. 260–268.

[18] A. R. D. Accardi and S. Chiarenza, "Musei digitali dell'architettura immaginata: un approccio integrato per la definizione di percorsi di conoscenza del patrimonio culturale Digital museums of the imagined architecture: an integrated approach," DISEGNARECON, vol. 9, 2016.

[19] P. Ferroni et al., "Risk Assessment for Venous Thromboembolism in Chemotherapy-Treated Ambulatory Cancer Patients," Med. Decis. Mak., vol. 37, no. 2, pp. 234–242, 2016.

[20] P. Ferroni, F. M. Zanzotto, N. Scarpato, S. Riondino, F. Guadagni, and M. Roselli, "Validation of a machine learning approach for venous thromboembolism risk prediction in oncology," Dis. Markers, vol. 2017, pp. 1–7, 2017.

[21] F. Guadagni et al., "RISK: A random optimization interactive system based on kernel learning for predicting breast cancer disease progression," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2017, vol. 10208 LNCS, pp. 189–196.

[22] F. Guadagni et al., "Personal and Sensitive Data in the e-Health-IoT Universe," in Internet of Things. IoT Infrastructures: Second International Summit, IoT 360{\textdegree} 2015, Rome, Italy, October 27-29, 2015, Revised Selected Papers, Part II, B. Mandler, J. Marquez-Barja, M. E. Mitre Campista, D. Cagá\vnová, H. Chaouchi, S. Zeadally, M. Badra, S. Giordano, M. Fazio, A. Somov, and R.-L. Vieriu, Eds. Cham: Springer International Publishing, 2016, pp. 504–514.

[23] A. Pieroni, N. Scarpato, and M. Brilli, "Industry 4.0 Revolution in Autonomous and Connected Vehicle A non-conventional approach to manage Big Data," J. Theor. Appl. Inf., vol. 96, no. 1, 2018.

[24] A. Pieroni, N. Scarpato, and M. Brilli, "Performance study in Autonomous and Connected Vehicle, a Industry 4.0 Issue", J. Theor. Appl. Inf., vol. 96, no. 2, 2018.

[25] M. Gerla, E. K. Lee, G. Pau, and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," in 2014 IEEE World Forum on Internet of Things, WF-IoT 2014, 2014, pp. 241–246.

[26] Iazeolla, G., Pieroni, A., D'Ambrogio, A., Gianni, D. A distributed approach to the simulation of inherently distributed systems (2010) Spring Simulation Multiconference 2010, SpringSim'10, art. no. 132.

[27] Iazeolla, G., Pieroni, A., D'Ambrogio, A., Gianni, D. A distributed approach to wireless system simulation (2010) 6th Advanced International Conference on Telecommunications, AICT 2010, art. no. 5489830, pp. 252-262.

[28] D'Ambrogio, A., Gianni, D., Iazeolla, G., Pieroni, A. Distributed simulation of complex systems by use of an HLA-transparent simulation language (2008) 2008 Asia Simulation Conference - 7th International Conference on System Simulation and Scientific Computing, ICSC 2008, art. no. 4675405, pp. 460-467.

[29] D'Ambrogio, A., Iazeolla, G., Pieroni, A., Gianni, D. A model transformation

approach for the development of HLA-based distributed simulation systems (2011) SIMULTECH 2011 - Proceedings of 1st International Conference on Simulation and Modeling Methodologies, Technologies and Applications, pp. 155-160.

[30] Gianni, D., D'Ambrogio, A., Iazeolla, G., Pieroni, A. Producing simulation sequences by use of a java-based generalized framework (2008) Proceedings - EMS 2008, European Modelling Symposium, 2nd UKSim European Symposium on Computer Modelling and Simulation, art. no. 4625266, pp. 171-176.

[31] D'Ambrogio, A., Gianni, D., Risco-Martín, J.L., Pieroni, A. A MDA-based approach for the development of DEVS/SOA simulations (2010) Spring Simulation Multiconference 2010, SpringSim'10, art. no. 142.