

Atta Method: A Nature Based Approach

Miguel Ecar¹

¹Software Engineering – Federal University of Pampa - UNIPAMPA
Alegrete – RS – Brazil

miguel@ecarsm.com

Abstract. *Traditional software engineering approaches were developed based on old manufacturing process, in term of roles, responsibilities, tasks and hierarchy. New approaches, known also as Agile approaches, are based on a fusion of these traditional approaches and the Agile Manifesto, which changes the perspective of main software process objectives. This paper proposes a new software engineering method called Atta, based on a nature process. Furthermore this paper explore the main current software approaches and contrast them with the proposed one. The result is a new software engineering method, which has bases on nature and brings a new vision of organizational processes, responsibilities, roles and task division. Nonetheless this method needs to be validated.*

1. Introduction

This paper presents a new proposal of agile software development method. Atta method is inspired in a nature existent society. Atta sexdens is a type of Leaf Cutter ant genus, which are among the most complex found in all ants [Wilson 1980].

This idea came from the need of a more structured hierarchy, specially when new members are integrated into the team. But even adding some more structured organization and some roles segmentation, is still lightweight, having basis on the agile manifesto and avoiding plan-based approaches.

The inspiration is group of insects to create and propose a new vision for agile software development. The Agile Manifesto was the start point to make a link with the knowledge about these insects organization.

Atta method, is composed by, roles, events, tasks and artifacts, all of them inspired in ants organization and translated to software engineering. It is classified as agile method, due to its principal basis in terms of software development was the Agile Manifesto which have four values, they are (1) Individuals and interactions over processes and tools; (2) Working software over comprehensive documentation; (3) Customer collaboration over contract negotiation; (4) Responding to change over following a plan [Beck et al. 2001].

Taking this into account, Atta method is an approach based on an existent society in nature which, has as common goal to survive and to increase with constant interaction among each member (value 1). They work to find inputs to farm their own food (value 2). They collaborate with the queen, and each caste inside the colony has its own tasks and responsibilities (value 3). Moreover, they adapt themselves to climate changes and attackers, and keep working for colony growth (value 4).

There are several approaches that are called agile methods, which are considered lightweight methods in that they are characterized by short, iterative development cycles, self-organizing teams, simpler designs, code refactoring, test-driven development, frequent customer involvement [Abran et al. 2004]. Agile methods are based on the notion of incremental development and delivery. They share a set of principles, based on the agile manifesto, and so have much in common [Sommerville 2010].

The rest of the paper is organized as follows, section 2 contains the background about leaf cutter ants. Section 3 detail how Atta method is composed. Finally, section 4 presents conclusions and future work.

2. Background

Leaf Cutter ants are among the most complex found in all ants, can be expected to provide an estimate of the upper limits of colonial organization in these insects [Wilson 1980]. They are one of the most important groups of insects because they attack plants intensely and constantly at any development stage [ZANETTI et al. 2002]. They are called leaf cutter due to main task is to cutter leaf peaces and take to the colony, these leaf peaces are used to grow a specific fungus which leaf cutter ants feed on [LOECK et al. 2001]. Due to its efficiency in leaf cutting, they are considered among the more important farming pests in Latin America [de Abreu and Delabie 1986, Hernández and Jaffé 1995].

The organization of these ants is composed by 4 castes. The figure 2 shows the organizational structure.

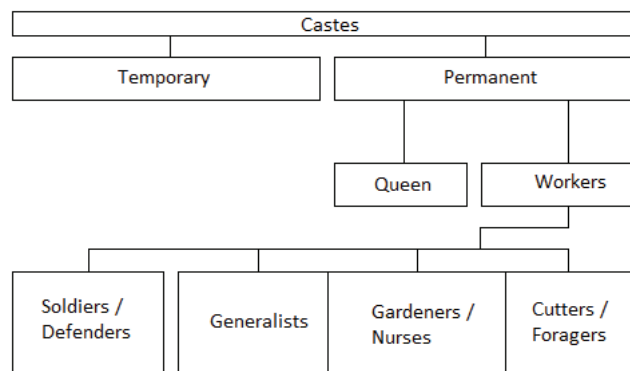


Figure 1. Atta Castes. adapted from [ZANETTI et al. 2002]

- **Gardeners and Nurses:** workers specialized on the care of fungus, final treatment and implantation of substrate particles, brood care, and parasol riding.
- **Generalists:** workers that perform a relatively wide variety of tasks, including the degradation of vegetation prior to its incorporation into the garden, major reconstruction of the fungus combs, disposal of refuse, transport of other workers and queen care.
- **Foragers or cutters:** workers that explore, recruit to new vegetation, cut and retrieve the vegetation, and excavate the nest.
- **Defenders or soldier:** workers that protect colony from invaders and help cutters.

3. Atta Method

Atta method emerged from the need to more structure in a agile software development environment. It is composed by roles, events, tasks and artifacts.

3.1. Roles

Atta Method is composed of five roles, they are: **Queen, Soldier, Cutter Dev. Team, Generalist Dev. Team and Quality Assessment Dev. Team** as shown in table 1

- **Queen:** The Queen is a single person, which is the owner of the business. This person is not part of development team in other words, the Queen do not code but its only responsibility is to feed the product.
- **Soldier Team:** The Soldiers must correspond for about 5% of the whole Atta Team. This team has mainly four responsibilities, coaching, consulting, management and also development. This team must be composed by specialists. Each Atta Team should decide what specialties are demand, for example, DBA, UX Designer, Manager, Tester, Measurer, etc. It is important to note that each specialty must be a single representative in the Soldier Team.
- **Cutter Development Team:** The Cutters is composed of junior developers. The team correspond for about 15% of Atta Team. Its responsibility is to develop a prototype, more precisely an High Quality/Fidelity Functional Prototype which follows the DoD”.
- **Generalist Development Team:** Generalists are responsible for make sure that the cutter team is following the DoD’. They correspond of about 30% of Atta Team and it is composed of junior developers and trainees, the team should test, white test script and detailed specification to be followed by QA Team.
- **Quality Assessment Team:** The QA Team should correspond for about 60% of Atta Team and must be filled with senior developers. They are responsible to take the prototype delivered by the Cutter Team and raise it to DoD”, following the detailed specification delivered by Generalist Team. Moreover, they are responsible for integration tests and make sure that the build keeps consistent with the new increment.

Table 1. Atta Team Roles

Role	Professional	%
Atta Queen	Business Man	*
Atta Soldiers	Specialists	5%
Atta Cutters	Junior Developers	15%
Atta Generalists	Junior Dev and Trainees	30%
Atta QA’s	Senior Developers	60%

*Single Person

3.2. WorkFlow

Atta workflow is presented in figure 3.2. The process starts with the Queen, which is responsible for maintain the product high level requirement specification. It is divided in

small increments that are taken by the Cutters team which develop the increment reaching the DoD'.

After that, the Cutters output is a high quality functional prototype that is taken by Generalists who develop test scripts and detailed requirement specification which is also part of DoD'. This step is iterative and go back to Cutters if specification detail is missing. After that, the responsibility goes to QA team, which increase the prototype developed by Cutters following detailed specification until reach the tests written by Generalist. This is also an iterative step which round in QA team until reaches DoD''. The out put from QA team is the increment added to the whole project.

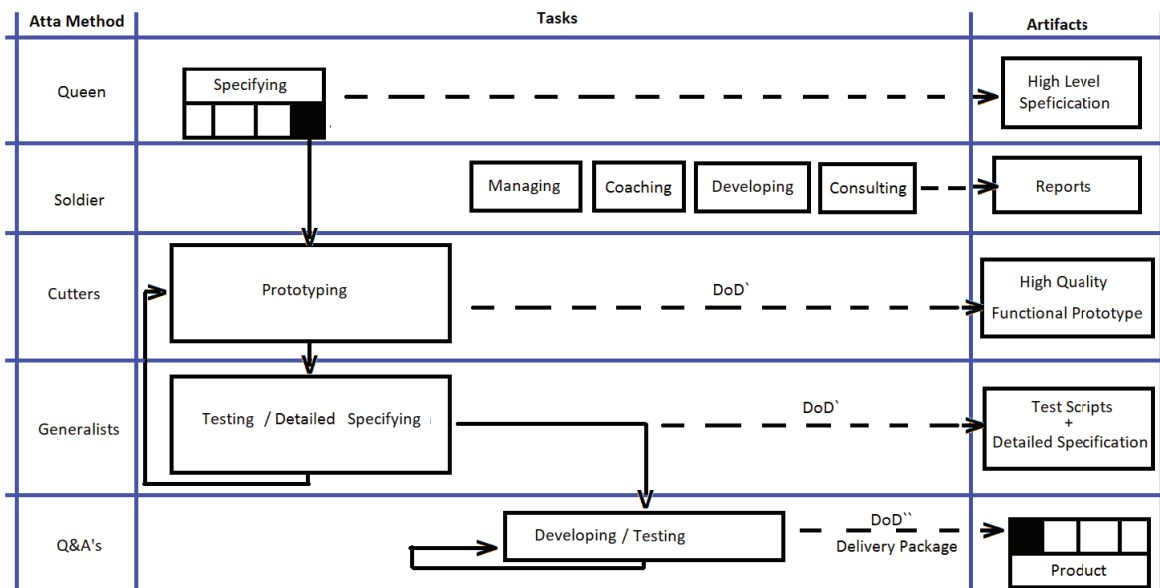


Figure 2. Atta Workflow

3.3. Events

Atta process has two main events, Cutting and Farming. Cutting, is a continuous event, this event is performed by Cutters and Generalists, that are continuously developing high quality functional prototypes. Farming is an event performed by Generalists and QA team, which is time-boxed. Protection is an event performed by Soldiers, which happen transversely in the process. The cycle is shown in figure 3.3.

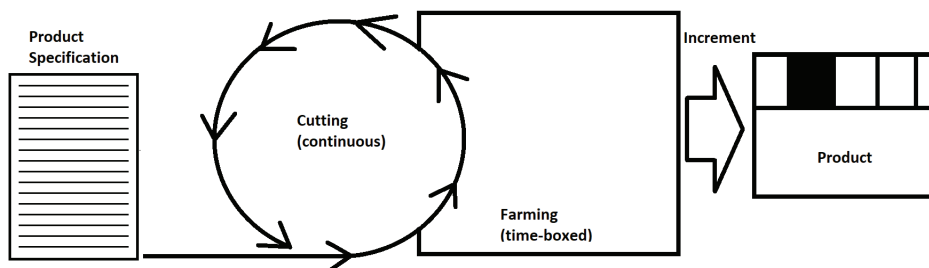


Figure 3. Atta Events Cycle

3.3.1. Cutting

The Cutting event is performed by Cutters and Generalists. It is classified as a continuous event and it is divided in two parts.

The first part a cutter team member (or simply cutter) takes a single part of requirements pile and starts its prototyping. This single part is called Leaf Clipping and it is characterized as the smallest part with value for the user. Moreover, a cutter must be able to prototype the increment by itself.

The second part, is performed by a Generalist. which should write detailed specification, the meaning of “detailed” is in DoD’, and test scripts.

It is important to note that cutters and generalists should work together, the one responsibility complement the other one responsibility. Thus, together both team members must reach the DoD’. Once, it is done, cutters and generalists star the cycle again, taking a new leaf clipping.

It is important to always have increments in DoD’ tag to supply at least one next Farming.

3.3.2. Farming

The Farming event is performed by QA team. It is classified as time-boxed event, in other words, it has a defined scope and should start and finish in a previously defined time. This time-box is calculated in the project begin and should be agreed by everyone, following the idea of “sprint” in scrum [Schwaber and Sutherland 2011]. The farming scope if calculated based on the amount of software that QA team manage to take from DoD’ until reach DoD” in the time-box.

Furthermore, DoD” should contemplate specification for a high quality increment and its integration to the rest of the system. Thus, the delivery at the end of farming is increments added to the whole software.

3.3.3. Protection

Protection is an event performed by the Soldiers team. Soldiers should ensure that the project is healthy, so they can act in any event performing tasks, such as, consulting, managing, coaching, measuring and etc, according to its specialty.

3.4. Artifacts

Atta method has a different number of artifacts. They are, definition of done, high level specification, reports, high quality prototype, test scripts, detailed specification, increment and incremented software.

The term “Definition of Done” was taken from Scrum glossary. When a Product Backlog item or an Increment is described as “Done”, everyone must understand what “Done” means. Although this varies significantly per Scrum Team, members must have

a shared understanding of what it means for work to be complete, to ensure transparency [Schwaber and Sutherland 2011].

In Atta Method, there are two DoD's, the one called DoD' (DoD one) that is defined to be the reach point for cutters and generalists, in terms of prototyping, test scripts and detailed requirements. The other one, is called DoD'' (DoD two) the is the reach point for QA team and should contemplate the minimum specification for increment integration in whole project.

High level specification is done by the Queen, which is the business person, which knows exactly what the software should do and how it should be done. It is indicated to use, high level specification writing techniques, such as, user story, epics, wire frames, or event simple plain text.

Reports are made by soldiers, according to its specialty, they should have in mind that a report is made when it aggregate value to Atta team as a whole. So, measurement analysis, controlling charts, manuals and etc, are performed by soldiers to make the process more transparent.

High quality prototype is expected from cutters and it should agree with DoD'. test scripts and detailed specification, are expected from generalists, they should also agree with DoD' and detailed specification must be done, also, when it is valuable for the whole Atta team, any writing technique may be used since, it is agreed with all members.

The increment and incremented software are expected from QA team at the end of farming, They should agree with DoD'' and are in fact the software expected by the Queen and final users.

3.5. Comparative Analysis

The Table 2 shows a comparative analysis among Atta and other agile approaches. The agile approaches compared are eXtreme Programming (XP) [Maurer and Martel 2002], Scrum [Schwaber and Sutherland 2011], Feature-Driven Development (FDD) [Palmer and Felsing 2001], Adaptive Software Development (ASD) [Highsmith 2013], Dynamic Systems Development Method (DSDM) [Stapleton 1997].

Based on Table 2, it is possible to note that Atta method has significant differences in terms of project size, team size, development style, environment and business culture, from other agile approaches.

4. Conclusion and Future Work

This paper presents a new software development method, called Atta method. It is inspired in a nature process, the leaf cutter ants organization.

The basis for this method development was the need for a more structured software development process. Current agile approaches left lacks of segmentation, and it may be difficult to implement in an existent company with huge number of developers.

Thereby, the raised hypothesis is that Atta method may feet in this type of organization bringing a segmented environment at the same time as an agile culture and thinking.

Table 2. Agile Approaches Comparative Analysis.

Criteria	XP	Scrum	FDD	ASD	DSDM	Atta
Scope Project Size	Small, medium	Small, medium, and scalable for large	Small, medium, and large (business projects/applications)	Large and Complex projects	Small and large projects (Business Applications)	Medium, Large and Complex projects
Team Size	<10	<10 and multiple teams	No limit, scalable from small to large teams	Not mentioned	Minimum 2 and Maximum 6 (Multiple teams)	No limit, scalable from small to large teams
Dev. Style	Iterative, rapid	Iterative, rapid	Iterative design and construction	Iterative and rapid development, distributed development	Iterative, rapid development and cooperative	Iterative, rapid development and cooperative
Physical Environment	Co-located teams and distributed teams (limited-interaction)	Not specified	Not specified	Co-located teams and distributed teams	Not mentioned	Co-located teams and distributed teams
Business Culture	Collaborative and cooperative	Not specified	Not specified	Not specified	Collaborative and cooperative	Collaborative and cooperative

*Adapted from [Qumer and Henderson-Sellers 2008]

Next step for this work is an empirical validation to serve as concept proof and refinement of this proposal. The plan is expose the Atta method idea and get feedback from community. It should be noted that Atta method is currently only an idea, but with refinement and enrichment, it may have a good contribution for agile software community.

References

- Abran, A., Moore, J. W., Bourque, P., Dupuis, R., and Tripp, L. L. (2004). *Guide to the software engineering body of knowledge: 2004 version SWEBOK*. IEEE Computer Society.
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., et al. (2001). The agile manifesto.
- de Abreu, J. and Delabie, J. (1986). Controle das formigas cortadeiras em plantios de cacau. *Revista Theobroma*, 16:4.
- Hernández, J. V. and Jaffé, K. (1995). Dano econômico causado por populações de formigas atta laevigata (f. smith) em plantações de pinus caribaea mor. e elementos para o manejo da praga. *Anais da Sociedade Entomológica do Brasil*, 24(2):287–298.
- Highsmith, J. (2013). *Adaptive software development: a collaborative approach to managing complex systems*. Addison-Wesley.
- LOECK, A. E., Grützmacher, D., and Storch, G. (2001). Distribuição geográfica de *atta sexdens piriventris santschi*, 1919, nas principais regiões agropecuárias do estado do rio grande do sul. *Current Agricultural Science and Technology*, 7(1).
- Maurer, F. and Martel, S. (2002). Extreme programming. rapid development for web-based applications. *IEEE Internet computing*, 6(1):86–90.
- Palmer, S. R. and Felsing, M. (2001). *A practical guide to feature-driven development*. Pearson Education.
- Qumer, A. and Henderson-Sellers, B. (2008). An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Information and software technology*, 50(4):280–295.
- Schwaber, K. and Sutherland, J. (2011). The scrum guide. *Scrum Alliance*, 21.
- Sommerville, I. (2010). *Software Engineering*. Addison-Wesley Publishing Company, USA, 9th edition.
- Stapleton, J. (1997). *DSDM, dynamic systems development method: the method in practice*. Cambridge University Press.
- Wilson, E. O. (1980). Caste and division of labor in leaf-cutter ants (hymenoptera: Formicidae: Atta). *Behavioral ecology and sociobiology*, 7(2):143–156.
- ZANETTI, R., Carvalho, G. A., Santos, A., Souza-Silva, A., and Godoy, M. S. (2002). Manejo integrado de formigas cortadeiras. *Lavras: UFLA*, page 16.