

## Explorando o cenário das Metodologias de Engenharia de Software Orientado a Agentes

Eduardo Augusto Ferreira da Silva<sup>1</sup>, Heder Dorneles Soares<sup>1</sup>, Rafael Sampaio Rocha Machado<sup>1</sup>

<sup>1</sup>Instituto de Computação – Universidade Federal Fluminense (UFF)  
Niterói – RJ – Brasil

{eferreira, hdorneles, rmachado}@ic.uff.br

**Abstract.** *The scenario methodologies of traditional software engineering is agreed and widespread. However, in the context of multiagent systems, yet to explore the field of agent-oriented methodologies because of the particularities. The objectives of this work are: (i) present and discuss the definition of what is and what is a methodology for software engineering-oriented agents, (ii) discuss the standards adopted by the methodologies ADELFE, Gaia, Message and MaSE; (iii) to compare these methods with the criteria of the life cycle, iterative, traceability and focus.*

**Resumo.** *O cenário de metodologias de engenharia de software tradicionais é convencionalizado e difundido. Entretanto, no contexto de Sistemas Multiagentes, ainda há de se explorar no campo das metodologias orientadas a agente devido as suas particularidades. Os objetivos deste trabalho são: (i) apresentar e discutir a definição do que é e no que consiste uma metodologia de engenharia de software orientado a agentes, (ii) discorrer sobre os padrões adotados pelas metodologias ADELFE, Gaia, Message e MaSE e (iii) comparar estas metodologias com os critérios de ciclo de vida, iteratividade, rastreabilidade e foco.*

### 1. Introdução

Independente da aplicação de metodologias, tanto para sistemas orientado a objetos quanto para sistemas orientados a agentes, é de suma importância utilizar a abordagem adequada para direcionar o desenvolvimento de softwares. Gerentes de projetos, desenvolvedores, arquitetos de software, engenheiros de software, analistas de testes e todos os demais profissionais envolvidos em projetos de software podem ter seus papéis e responsabilidades melhores definidos sobre as atividades exigidas para realização do desenvolvimento e entrega do sistema. Além do que, o uso de metodologia apoiando no desenvolvimento de software, através do uso de boas práticas, técnicas e ferramentas, tem o objetivo de proporcionar ao produto de software final uma gestão de escopo, tempo, custo, qualidade e recurso. Metodologias são os resultados de transferência de conhecimento da engenharia de software. No cenário de sistemas multiagentes [Iglesias et. al. 1999], as metodologias de desenvolvimento de software orientado a agentes não possuem a mesma maturidade como a abordagem orientada ao objeto.

O escopo deste artigo é: (i) apresentar e discutir a definição do que é e no que consiste uma metodologia de engenharia de software orientado a agentes, (ii) discorrer sobre os padrões adotados pelas metodologias ADELFE, Gaia, Message e MaSE e (iii) comparar estas metodologias com os critérios de ciclo de vida, iteratividade,

rastreadabilidade e foco. Sendo assim, será apresentada uma perspectiva sobre o rumo das metodologias orientadas a agente.

## 2. Conceituando Metodologias de Software

Por definição, a aplicação de uma engenharia de software compreende em utilização de sistemática e disciplina em uma abordagem quantificável para operação e manutenção de software. Portanto, engenharia de software e metodologias consiste no estudo de como são aplicadas estas abordagens.

O termo *Software engineering* (do português engenharia de software) surgiu inicialmente em 1968 no nome de uma conferência patrocinada por North Atlantic Treaty Organization (*NATO*). O foco desta conferência foi dado pelos diversos problemas que afetam a indústria de software: atraso das entregas, complexidade do produto final, custo de projeto. [Jorge J. Gomez-Sanz et. al. 2011]

A utilização de engenharia de software e metodologias engloba questões essenciais, em especial: competências técnicas e humanísticas. As competências técnicas são requisitos fundamentais para os analistas, como por exemplo, conhecimento da linguagem de programação mais adequada para a complexidade da aplicação a ser desenvolvidas ou então o próprio conhecimento de boas práticas, ferramentas e técnicas a serem adotadas nas atividades de desenvolvimento do sistema [Jorge J. Gomez-Sanz et. al. 2011].

### 2.1. SWEBOK

O conhecimento de engenharia de software é definido e especificado no *Software Engineering Book of Knowledge* (SWEBOK). O foco é no conhecimento essencial que suporte a seleção da tecnologia apropriada, no tempo e na circunstância apropriados.

No SWEBOK são descritas onze áreas de conhecimento. Porém, a área que deve ser levada em conta pela relevância no contexto de sistemas multiagentes é a área de “Ferramentas e Métodos de Engenharia de Software”. Segue abaixo as citações sobre o conceito de ferramentas, métodos e suas relações:

“Ferramentas são frequentemente projetadas para suportar métodos de engenharia de software particulares, reduzindo qualquer carga administrativa associada com aplicação dos métodos manuais. Como métodos de engenharia de software, ferramentas pretendem fazer a engenharia de software mais sistemática e variam no escopo de suporte individual de tarefas para abranger o ciclo de vida completo.” [IEEE, SWEBOK 2004]

“Métodos de engenharia de software impõem estrutura na atividade de engenharia de software com o objetivo de fazer a atividade sistemática e ultimamente mais sucedida. Métodos usualmente provem uma notação e vocabulário, procedimentos para desempenharem tarefas identificáveis e princípios básicos para checarem tanto processo quanto produto. Métodos variam amplamente no escopo, de uma única fase de ciclo de vida para completar o ciclo de vida. A ênfase nesta área de conhecimento é métodos de engenharia de software abrangendo múltiplas fases de ciclo de vida, desde métodos de fases específicas abordadas por outras áreas do conhecimento.” [IEEE, SWEBOK 2004]

Considerando as definições expostas, é intrínseca e tênue a relação de métodos e ferramentas. Processos, notações, técnicas e linguagens a serem utilizadas são desafios e objetos de estudo para a modelagem de métodos e ferramentas.

### **3. A abordagem da orientação a agentes**

No contexto da Engenharia de Software Orientada a Agentes pode-se categorizar o método heurístico cujo sistema é interpretado como uma coleção de agentes. O método deve ser entendido como bem sucedido ao torná-lo sistemático e cíclico.

O método também é possível fornecer notação, vocabulário, procedimentos para desempenharem tarefas identificáveis e princípios básicos para checagem tanto de processos quanto de produtos [Jorge J. Gomez-Sanz et. al. 2011]. Ou seja, os conceitos básicos e o que se esperar de metodologias e ferramentas pelo SWEBOK também pode ser utilizado no universo de Engenharia de Software Orientada a Agentes.

Há dois grandes grupos de propostas de metodologias. Uma metodologia que primeiro tenta construir uma notação e vocabulário para direcionar algumas fases de desenvolvimento específicas. Na outra, notação e vocabulário são contempladas a ferramentas que permitem andar com a análise e design para implementação. [Jorge J. Gomez-Sanz et. al. 2011].

Apesar da distinção, esses dois grupos estão se convergindo. O que se espera de metodologias candidatas são que tenham ferramentas de suporte (similar a existente na metodologia ZEUS) e se atentem mais ao ciclo de vida do software. Metodologias como INGeniAS, PASSI, ADELFE ou Prometheus crescem pois são caracterizadas por obter maior padronização, definição de fases de desenvolvimento, coleta de requisitos, implementação e testes.

### **4. Metodologias orientadas a agentes**

As metodologias precisam ser usadas continuamente para evoluírem de maneira progressiva. Sendo assim, a evolução poderá direcionar novas atividades de desenvolvimento e incrementar o escopo da metodologia. Há estudos anteriores [Dam 2004] que já realizaram comparações sobre as metodologias.

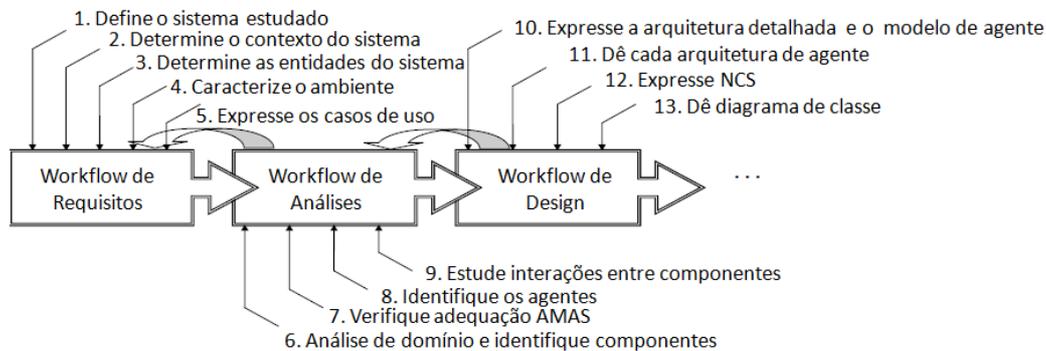
As particularidades de sistema multiagentes se diferem das outras devido às características específicas de definição de agente: comunicação entre agentes, normas, recompensa, proibição e raciocínio. Nos próximos tópicos, serão brevemente abordadas algumas das metodologias orientadas a agentes.

#### **4.1. ADELFE**

ADELFE é comparado ao Processo Unificado da Rational (RUP), porém dedicada a engenharia de software para sistemas multiagente adaptativos. É comparado ao RUP por definir primeiro três núcleos de workflows. Durante a fase de requisitos, o ambiente do sistema estudado deve ser definido e caracterizado. Então, na fase de análise, o engenheiro é guiado para decidir o uso de tecnologia de multiagente adaptativa e a identificação de agentes junto ao sistema e aos modelos de ambiente. Finalmente, o workflow de projeto do ADELFE deve fornecer o modelo de agente cooperadora e ajudar a desenvolver a definição de comportamento de agentes.

ADELFE é baseado em metodologias orientada a objeto seguindo o Processo Unificado da Rational (RUP) e usa notações UML e AUML. Alguns passos foram

adicionados nos workflows clássicos para serem específicos para adaptação de sistemas multiagentes. Isto não é uma metodologia geral como a GAIA ou TROPOS, mas tem um nicho de aplicações interessantes que requerem design de sistemas multiagentes adaptativos usando a teoria AMAS. ADELFE segue o processo inteiro de engenharia de software como MESSAGE, PASSI e TROPOS. [Carole Bernon et. al. 2002]. Veja abaixo a ilustração contendo as atividades de cada *workflow* e seu caráter cíclico:

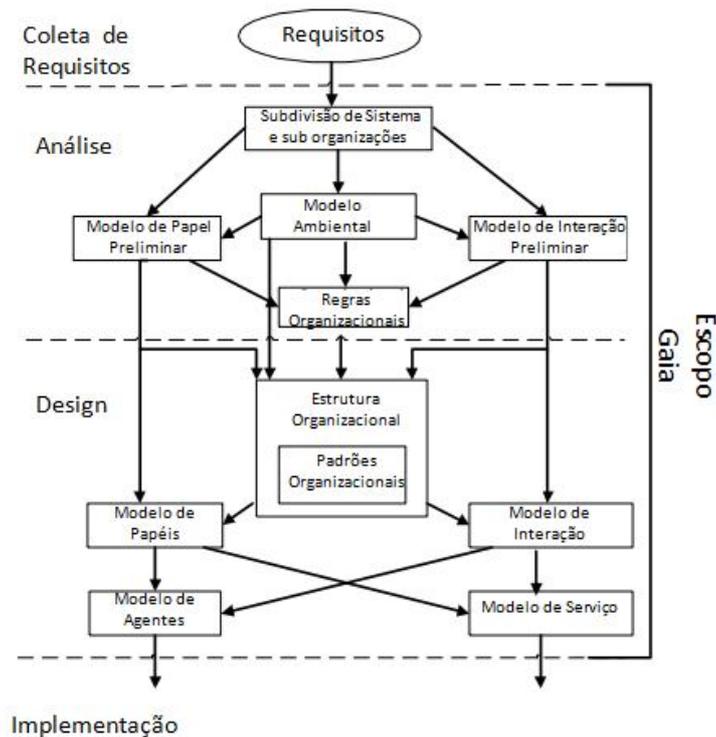


**Figura 6 Workflows ADELFE [Adaptado de Bernon et. al. 2002]**

A metodologia ADELFE é direcionada para sistemas multiagentes com ambientes e características de sistemas adaptativos, com comportamentos dinâmicos. O que caracteriza esta metodologia de maneira interessante é a similaridade com o RUP e uso dos três workflows no seu ciclo de vida: requisito, análise e projeto (design). Em cada workflow, similar ao RUP, é possível revisitar cada etapa e atividade, sempre priorizando a atualização e definição mais completa do sistema.

## 4.2. Gaia

A metodologia GAIA é basicamente dividida em duas definições: Análise e Design. Na etapa de análise é feito: a decomposição de organização em sub organizações, modelo de ambiente, modelo de papéis preliminares, modelo de interação preliminar e regras da organização. A seguir a ilustração contendo o escopo da GAIA:



**Figura 7 Modelo de Gaia e suas Relações [Zambonelli et al., 2003]**

Na etapa de design as definições previamente concebidas na análise são consolidadas no que se prevê em: topologia e controle da estrutura na estrutura organizacional, modelo final de papéis, modelo final de interação, modelo de agente e modelo de serviço.

A metodologia Gaia é uma das mais utilizadas devido a sua simplicidade e especificação razoavelmente definida. Além da versão inicial, também houve evolução, sendo nomeada para a versão 2: Em particular, em Gaia v.2, em adição às funções e protocolos, o ambiente no qual um sistema de multi agente está imerso é escolhido para uma análise primária e abstração de design. [Cernuzzi et al., 2004]

Diferente de outras metodologias como a ADELFE e TROPOS, a metodologia GAIA não aborda a fase de requisito. A GAIA não apresenta técnicas de modelagem particulares ou sistemas de notações específicos. Outra fase importante não contemplada no escopo desta metodologia é a implementação, não caracterizando uma metodologia completa em percorrer todo o ciclo de vida de desenvolvimento de um sistema multiagente.

### 4.3. Message

MESSAGE baseia-se na definição da metodologia no uso de meta modelos como formalismo na especificação. Linguagens de meta modelagens fazem possível a extensão do núcleo de especificação pelo uso de mecanismo de geração de construção. Também, instâncias dos meta modelos (ex: os modelos que descrevem MAS), se opõem para maioria dos formalismos, podendo ser detalhado parcialmente e refinado junto de etapas sucessivas (similarmente para UML, onde diagramas de classes não precisam ser completos na iniciação, mas podem crescer em detalhes durante o desenvolvimento. [Jorge et. al. 2001]

Uma característica relevante da metodologia MESSAGE é a adoção da base do RUP, como adotado na ADELFE. A evolução e maturidade da metodologia MESSAGE podem ser comprovadas pelo manual técnico [Caire et. al. 2002] de especificação da metodologia. Muitas atividades podem ser um argumento para desacreditar a metodologia, mas em fato nós devemos considerar que atividades são definidas para cada visão do sistema (há cinco em MESSAGE: organização, agente, tarefas/objetivos, interações e domínios) e essas visões são normalmente desenvolvidas em paralelo (devido a suas relações). Também, engenheiros de softwares rodeiam a aplicação de algumas atividades quando deparam com domínios de aplicação simples. [Jorge et. al. 2001]

#### 4.4. MaSE

A Engenharia de Sistema Multi Agente - MaSE metodologia pega uma especificação inicial e produz um conjunto de documentos de design formais em um estilo baseado graficamente.

O foco primário da MaSE é guiar o designer junto ao ciclo de vida de software por uma prosa de especificação para uma implementação de sistema de agente. MaSE é independente de uma arquitetura de sistema multiagente particular, arquitetura de agente, linguagem de programação ou sistema de troca de mensagem. Um sistema projetado em MaSE pode ser implementado em diversas maneiras diferentes de um mesmo design. MaSE também oferece e habilita o acompanhamento de mudanças através de processo. Todo objeto de design pode ser traçado posteriormente ou anteriormente por diferentes fases da metodologia junto dos seus correspondentes de construção [M. Wood e S. DeLoach 2001].

A metodologia MaSE pode ser considerada uma metodologia tradicional pelas definições de modelos e fases. A seguir está a visão geral da metodologia MaSE e seus modelos:

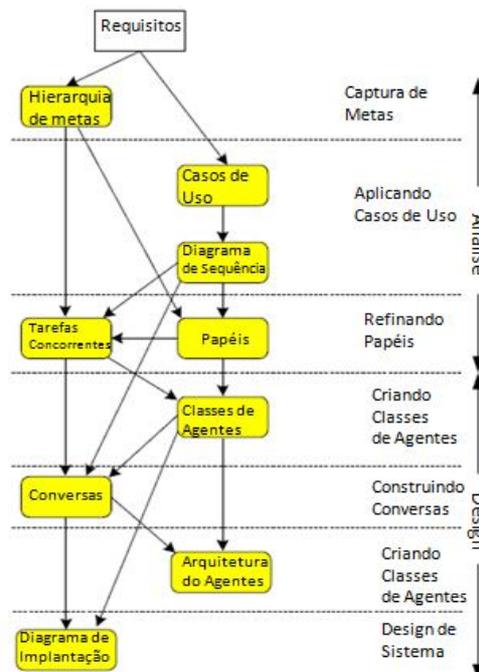


Figura 8 A Metodologia MaSE [Adaptado de M. Wood e S. DeLoach 2001]

## 6. Resultado da comparação das metodologias

Baseado no estudo apresentado, segue abaixo (quadro 1) a comparação das metodologia orientada a agentes pelos seguintes critérios: (a) Ciclo de vida, (b) iteratividade, (c) rastreabilidade e (d) foco.

**Quadro 3 Comparação de Metodologia por Critérios**

<b>Metodologia</b> <b>Critério</b>	<b>ADELFE</b>	<b>Gaia</b>	<b>Message</b>	<b>MaSE</b>
<b>Ciclo de vida</b> (refere-se ao ciclo de vida em fases da metodologia)	Requisito, Análise, Design, Implementação e Teste.	Análise, Design	Coleta de Requisito, Análise, Design	Análise e Design
<b>Iteratividade</b> (indica se há ciclos de iteratividade na metodologia)	Sim. Baseada no RUP.	Não especificado.	Sim. Baseada no RUP.	Sim. Baseada no RUP.
<b>Rastreabilidade</b> (refere-se a relação de comunicação entre os artefatos)	Sim. Há evolução dos artefatos por iteração.	Sim. Interrelações entre modelos.	Sim. Há evolução dos artefatos por iteração.	Sim. Interrelações entre modelos.
<b>Foco</b> (refere-se ao conjunto de modelagem contemplados e priorizados pela metodologia)	Requisito, Análise, Design,	Análise e Design	Análise, Design	Análise e Design

## 7. Conclusão

Com base no estudo e nas comparações das metodologias orientadas a agentes abordados neste trabalho, espera-se fomentar discussão na comunidade de Engenharia de Software Orientada a Agentes. Apesar da heterogeneidade presente nestas metodologias comparadas, há similaridade nos resultados dos critérios avaliados.

Há muito a se explorar na área de engenharia de software orientada a multiagentes, inclusive o aperfeiçoamento das metodologias existentes na definição e melhoramento contínuo de processos, técnicas, métodos e ferramentas. Para progressão da área de metodologias de engenharia de software orientada a agentes, há uma forte demanda de melhoria geral em ferramentas de suporte e aumento do número de desenvolvedores para avaliar as possibilidades de cada metodologia.

## Referências

- Caire, G., Leal, F., Chainho, P., Evans, R., Garijo, F., Gomez-Sanz, J. J., Pavon, J., Kerney, P., Stark, J., and Massonet, P. Eurescom P907: MESSAGE - Methodology for Engineering Systems of Software Agents. <http://www.eurescom.de/public/projects/P900-series/p907/default.asp> . 2002.
- Carole Bernon , Marie-Pierre Gleizes , Sylvain Peyruqueou , Gauthier Picard, ADELFE: a methodology for adaptive multi-agent systems engineering, Proceedings of the 3rd international conference on Engineering societies in the agents world III, September 16-17, 2002, Madrid, Spain

- Cernuzzi, L., Juanand, T., Sterling, L., Zambonelli, F.: The Gaia Methodology: Basic Concepts and Extensions. In: Methodologies and Software Engineering for Agent Systems. Kluwer, Dordrecht (2004)
- Dam, Khanh Hoa, and Michael Winikoff. "Comparing agent-oriented methodologies." Agent-Oriented Information Systems. Springer Berlin Heidelberg, 2004.
- IEEE. Guide to the Software Engineering Body of Knowledge (SWEBOK). 2004 Version.
- Iglesias, Carlos A., Mercedes Garijo, and José C. González. "A survey of agent-oriented methodologies." Intelligent Agents V: Agents Theories, Architectures, and Languages. Springer Berlin Heidelberg, 1999. 317-330.
- Jorge J. Gomez-Sanz, Ruben Fuentes-Fernández, Juan Pavon (2011), Understanding Agent Oriented Software Engineering Methodologies. AOSE 2011.
- Jorge J. Gómez-Sanz, Juan Pavón. Agent Oriented Software Engineering with MESSAGE. 2001.
- M. Wood, S. DeLoach. An Overview of the Multiagent Systems Engineering Methodology, In P. Ciancarini and M. Wooldridge, editors, Agent-Oriented Software Engineering - First International Workshop (AOSE), Limerick, Ireland, June 10, 2000. Lecture Notes in Computer Science. Vol. 1957, Springer Verlag, Berlin, 2001.
- Zambonelli, F., Jennings, N. R., and Wooldridge, M.J. (2003). Developing multiagent systems: The Gaia methodology. ACM Transactions on Software Engineering and Methodology, 12(3):417-470.