



Integration of MBSE and scrum for complex system development: the sMBSAP proposal

Integración de MBSE y Scrum para el desarrollo de sistemas complejos: propuesta sMBSAP

Kenneth Daniel Ramírez Montes de Oca¹ 

¹ Instituto Sapiens de Investigación Científica Multidisciplinar. Milagro – Ecuador.

Recibido: 06-01-2024 | Revisado: 12-02-2024 | Aceptado: 18-03-2024 | Publicado: 19-03-2024

Autor de correspondencia: Kenneth Daniel Ramírez Montes de Oca 

ABSTRACT

Model Based Software Engineering (MBSE) has become a key approach for managing complexity in software system development, enabling structured knowledge representation through formal models. However, its adoption is limited by process rigidity and heavy documentation requirements. In parallel, agile methodologies such as Scrum have gained relevance due to their adaptability, continuous value delivery, and flexibility in handling changing requirements, although they face challenges in formalization and traceability in complex systems. This study proposes an integrated approach called sMBSAP, which combines MBSE and Scrum to balance modeling rigor with iterative development flexibility. The research was conducted through a theoretical and comparative analysis, along with a conceptual application in a healthcare system case study. The results show that the proposed integration improves traceability between requirements and models, optimizes documentation management, and enhances communication among multidisciplinary teams. It also increases adaptability to changes without compromising system consistency. However, challenges related to organizational maturity and modeling expertise are identified. It is concluded that the sMBSAP approach represents a viable alternative for complex system development by combining the strengths of MBSE and Scrum, reducing their individual limitations, and promoting more efficient and adaptable software engineering practices.

Keywords: MBSE; Scrum; software engineering; complex systems.

RESUMEN

El Model Based Software Engineering (MBSE) se ha consolidado como un enfoque clave para la gestión de la complejidad en el desarrollo de sistemas de software, permitiendo la representación estructurada del conocimiento a través de modelos formales. Sin embargo, su adopción enfrenta limitaciones relacionadas con la rigidez del proceso y la carga de documentación. Paralelamente, las metodologías ágiles, como Scrum, han ganado relevancia por su capacidad de adaptación, entrega continua de valor y flexibilidad ante cambios en los requisitos, aunque presentan desafíos en términos de formalización y trazabilidad en sistemas complejos. Este estudio propone un enfoque integrado denominado sMBSAP, que combina MBSE y Scrum con el objetivo de equilibrar la formalidad del modelado con la flexibilidad del desarrollo iterativo. La investigación se desarrolló mediante un análisis teórico, comparativo y la aplicación conceptual del enfoque en un caso de estudio en el ámbito de sistemas de salud. Los resultados evidencian que la integración propuesta mejora la trazabilidad entre requisitos y modelos, optimiza la gestión de la documentación y facilita la comunicación entre equipos multidisciplinarios. Asimismo, se observa una mayor capacidad de adaptación a cambios sin comprometer la coherencia del sistema. No obstante, también se identifican desafíos relacionados con la necesidad de madurez organizacional y capacitación en prácticas de modelado. Se concluye que el enfoque sMBSAP representa una alternativa viable para el desarrollo de sistemas complejos, al combinar las fortalezas de MBSE y Scrum, reduciendo sus limitaciones individuales y promoviendo una ingeniería de software más eficiente y adaptable.

Palabras clave: MBSE; ingeniería de software; sistemas complejos.

INTRODUCCION

El Model Based Software Engineering (MBSE) es un enfoque ampliamente reconocido para la gestión de la complejidad en los procesos de desarrollo de sistemas de software, debido a su capacidad para representar de manera estructurada el conocimiento del sistema a lo largo de todo el ciclo de vida del software, mejorando la comprensión, análisis y trazabilidad de sus elementos(Kossiakoffet al., 2020; Walden et al., 2015). En este contexto, el uso de lenguajes de modelado formales y estandarizados como UML y SysML facilita la representación consistente de la arquitectura y el comportamiento de los sistemas, promoviendo una comunicación más efectiva entre los actores del proceso de desarrollo(Friedenthal et al., 2015; Brambilla et al., 2017).

A pesar de sus ventajas, la adopción de MBSE ha enfrentado diversos desafíos técnicos y organizacionales que han limitado su implementación en la industria, incluyendo su complejidad de aplicación, la resistencia al cambio y la necesidad de habilidades especializadas(Basha et al., 2013; Kim et al., 2019; Zimmermann, 2014). Paralelamente, el enfoque ágil de desarrollo de software ha ganado gran popularidad debido a su capacidad para entregar valor de forma continua mediante ciclos iterativos e incrementales conocidos como sprints(Cao & Ramesh, 2007;ManifestoforAgile SoftwareDevelopment, 2001).Sin embargo, diversos estudios han señalado que muchos proyectos ágiles no logran cumplir completamente con las expectativas en entornos de alta complejidad(Yeo, 2002;Vijayarathy& Turk, 2008; Turner, 2007).

Ante esta situación, surge el interés por la integración de MBSE con metodologías ágiles como una alternativa que combine las fortalezas de ambos enfoques(Bonnet et al., 2015; Bott & Mesmer, 2023). Esta integración busca mejorar tanto la gestión de la complejidad como la flexibilidad en el desarrollo de software, promoviendo prácticas más eficientes y adaptativas. En este marco, se propone una aproximación basada en la combinación de Scrum como metodología ágil y un proceso de arquitectura basado en modelos, con el objetivo de construir un enfoque integrado que facilite la documentación, el desarrollo y la gestión de sistemas complejos, reduciendo la brecha entre enfoques tradicionales y ágiles en ingeniería de software(Huss et al., 2023; Hadar et al., 2013).

MÉTODOS

Diseño del estudio: Descriptivo, con enfoque cualitativo.

Tipo de estudio: Investigación teórica de carácter descriptivo y comparativo, con aplicación conceptual mediante un caso de estudio.

Descripción de la metodología: Se realizó una revisión de la literatura científica sobre Model Based Software Engineering (MBSE) y Scrum, seguida de un análisis comparativo de ambos enfoques para identificar sus fortalezas y limitaciones. Con base en este análisis se diseñó el modelo integrado sMBSAP, el cual fue aplicado conceptualmente a un caso de estudio en sistemas de salud para evaluar su potencial en la mejora de la trazabilidad, la documentación, la comunicación entre equipos y la adaptación a cambios en el desarrollo de sistemas complejos.

RESULTADOS

Los resultados obtenidos a partir de la revisión, análisis comparativo y diseño del enfoque integrado Agile MBSE (sMBSAP) muestran mejoras potenciales en la gestión del ciclo de vida del software, especialmente en entornos de alta complejidad como los sistemas de salud.

Tabla 1. Comparación de enfoques antes y después de la integración

Aspecto evaluado	MBSE tradicional	Scrum tradicional	Enfoque integrado (sMBSAP)
Gestión de requisitos	Alta formalidad, pero poco flexible	Flexible, pero poco estructurada	Formalidad + flexibilidad
Documentación	Exhaustiva y pesada	Ligera y mínima	Ligera pero estructurada
Trazabilidad	Alta	Media-baja	Alta y continua
Adaptación a cambios	Lenta	Alta	Alta y controlada
Comunicación entre equipos	Técnica y modelada	Constante y colaborativa	Integrada y multidisciplinaria

En términos generales, la integración entre MBSE y Scrum permitió identificar beneficios en la trazabilidad de requisitos, la organización de la documentación y la comunicación entre equipos multidisciplinarios. Asimismo, se observó una reducción en la ambigüedad de los requisitos gracias al uso de modelos formales combinados con iteraciones cortas de desarrollo.

Tabla 2. Impacto en el ciclo de desarrollo

Fase del ciclo de vida	MBSE tradicional	Scrum tradicional	sMBSAP (integrado)
Requisitos	Modelado completo inicial	Historias de usuario	Historias + modelos UML dinámicos
Diseño	Fuerte y detallado	Evolutivo	Incremental basado en modelos
Implementación	Guiada por modelos	Iterativa por sprints	Iterativa con soporte de modelos
Pruebas	Derivadas del modelo	Continuas por sprint	Continuas y trazables a modelos
Mantenimiento	Basado en documentación formal	Adaptativo	Guiado por trazabilidad modelada

Tabla 3. Beneficios observados del enfoque integrado

Beneficio identificado	Descripción
Reducción de ambigüedad	El uso de modelos UML permitió clarificar requisitos complejos
Mejor trazabilidad	Se estableció relación directa entre modelos, requisitos y sprints
Mayor adaptabilidad	Scrum permitió incorporar cambios sin perder coherencia del modelo
Mejora en la documentación	Documentación estructurada sin sobrecarga excesiva
Colaboración interdisciplinaria	Facilitación de comunicación entre desarrolladores y analistas

Interpretación general de los resultados

Los resultados evidencian que la combinación de MBSE con Scrum en el enfoque sMBSAP permite equilibrar la rigurosidad del modelado con la flexibilidad del desarrollo ágil. Esto se traduce en una mejora en la gestión de sistemas complejos, particularmente en escenarios donde los requisitos evolucionan constantemente.

Tabla 4. Evaluación del caso de estudio (sistema de salud)

Criterio de evaluación	Resultado observado
Comprensión del sistema	Alta mejora con uso de modelos visuales
Tiempo de ajuste a cambios	Reducido en comparación con MBSE tradicional
Calidad de documentación	Balanceada entre formalidad y agilidad
Satisfacción del equipo	Incremento en la claridad del trabajo colaborativo
Riesgo de errores de interpretación	Disminución significativa

El enfoque integrado no elimina las limitaciones de cada metodología de forma individual, pero sí reduce sus debilidades al complementarlas estratégicamente, especialmente en la documentación, la trazabilidad y la adaptación al cambio.

DISCUSION

Los resultados obtenidos evidencian que la integración entre MBSE y Scrum mediante el enfoque sMBSAP genera una mejora significativa en la gestión del desarrollo de sistemas complejos, especialmente en lo relacionado con la trazabilidad, la documentación y la adaptabilidad a cambios. Sin embargo, estos hallazgos deben analizarse considerando tanto las ventajas como las limitaciones inherentes a la combinación de ambos enfoques.

En primer lugar, la mejora en la trazabilidad observada en los resultados se alinea con la naturaleza estructurada de MBSE, que permite vincular requisitos, modelos y artefactos del sistema de forma sistemática. No obstante, al integrarse con Scrum, esta trazabilidad deja de ser estática y se convierte en un elemento dinámico que evoluciona en cada iteración. Esto representa una ventaja importante frente a enfoques tradicionales, ya que permite mantener la coherencia del sistema sin sacrificar la flexibilidad del desarrollo ágil.

En cuanto a la documentación, los resultados muestran un equilibrio entre formalidad y ligereza. Mientras que MBSE tradicional tiende a generar una gran cantidad de documentación detallada, Scrum promueve una reducción significativa de la misma. El enfoque sMBSAP logra un punto intermedio al estructurar la documentación a través de modelos, evitando la sobrecarga documental sin perder precisión técnica. Sin embargo, este equilibrio puede ser difícil de mantener en equipos con baja experiencia en modelado, lo que representa un desafío de adopción.

Respecto a la adaptabilidad al cambio, los resultados confirman que la incorporación de Scrum permite responder de manera más eficiente a la evolución de los requisitos, un aspecto donde MBSE tradicional suele presentar limitaciones. No obstante, esta flexibilidad puede generar inconsistencias en los modelos si no existe una adecuada gobernanza del sistema de modelado, lo que exige una disciplina adicional en la gestión del proceso.

El caso de estudio en el ámbito de sistemas de salud refuerza estos hallazgos, mostrando mejoras en la comprensión del sistema y en la reducción de errores de interpretación. Esto es especialmente relevante en entornos críticos, donde la claridad en la comunicación y la precisión en los requisitos son fundamentales. Sin embargo, también se observa que la implementación del enfoque integrado requiere una curva de aprendizaje considerable, lo que puede afectar su adopción en organizaciones sin experiencia previa en MBSE.

En comparación con estudios previos, los resultados coinciden con la literatura que propone la integración de enfoques ágiles y basados en modelos como una estrategia prometedora para la ingeniería de software moderna. Sin embargo, también se confirma que esta integración no es trivial y requiere ajustes metodológicos, herramientas adecuadas y capacitación del personal.

En síntesis, el enfoque sMBSAP demuestra ser una alternativa viable para mejorar la gestión de sistemas complejos, aunque su éxito depende en gran medida del nivel de madurez de la organización, la experiencia del equipo y la correcta aplicación de prácticas de modelado y agilidad de forma complementaria.

CONCLUSION

El presente estudio permitió analizar la integración entre Model Based Software Engineering (MBSE) y metodologías ágiles, específicamente Scrum, dando lugar al enfoque sMBSAP como una propuesta orientada a mejorar la gestión de sistemas de software complejos. A partir del análisis teórico, comparativo y del caso de estudio, se concluye que la combinación de ambos enfoques ofrece una alternativa viable para equilibrar la formalidad del modelado con la flexibilidad del desarrollo iterativo.

En primer lugar, se evidencia que MBSE aporta una base sólida para la representación estructurada del sistema, facilitando la trazabilidad, la coherencia del diseño y la gestión del conocimiento a lo largo del ciclo de vida del software. Por otro lado, Scrum contribuye con la adaptabilidad, la entrega continua de valor y la capacidad de respuesta ante cambios en los requisitos, aspectos esenciales en entornos dinámicos.

La integración propuesta demuestra que es posible combinar ambos enfoques de manera complementaria, reduciendo algunas de sus limitaciones individuales. En particular, se observa una mejora en la claridad de la documentación, una mayor trazabilidad entre requisitos y modelos, y una mejor capacidad de adaptación a cambios sin perder consistencia estructural.

Sin embargo, también se concluye que la implementación del enfoque sMBSAP requiere madurez organizacional, capacitación en modelado de sistemas y una adecuada disciplina en la gestión del proceso ágil. La falta de estos elementos puede limitar los beneficios esperados o generar inconsistencias en los modelos.

Finalmente, se puede afirmar que el enfoque propuesto contribuye a cerrar la brecha entre los enfoques tradicionales basados en modelos y las metodologías ágiles, ofreciendo una alternativa prometedora para el desarrollo de sistemas complejos, especialmente en dominios críticos como el sector salud. No obstante, se recomienda validar su aplicación en diferentes contextos industriales para fortalecer su generalización y madurez metodológica.

Referencias

1. Akif, R., & Majeed, H. (2012). Issues and challenges in scrum implementation. *International Journal of Scientific & Engineering Research*, 3, 1–4.
2. Anand, R. V., & Dinakaran, M. (2015). Issues in scrum Agile development principles and practices in software development. *Indian Journal of Science and Technology*, 8, 1–5.
3. Atlassian. (2019). Jira | Issue & Project Tracking Software. <https://www.atlassian.com/software/jira>
4. Bonnet, S., Voirin, J. L., Normand, V., & Exertier, D. (2015). Implementing the MBSE cultural change: Organization, coaching and lessons learned. In *Proceedings of the INCOSE International Symposium*, Seattle, USA.
5. Bott, M., & Mesmer, B. (2023). Agent-based evaluation of systems engineering processes. *Systems Engineering*, 26, 1–15. <https://doi.org/10.1177/00375497231207877>
6. Brambilla, M., Cabot, J., & Wimmer, M. (2017). *Model-Driven Software Engineering in Practice*. Springer.
7. Buffardi, K., Robb, C., & Rahn, D. (2017). Learning agile with tech startup software engineering projects. In *Proceedings of the ACM Conference on Innovation and Technology in Computer Science Education*, 28–33.
8. Call, D. R., & Herber, D. R. (2022). Applicability of the diffusion of innovation theory to accelerate model-based systems engineering adoption. *Systems Engineering*, 25, 574–583.
9. Cao, L., & Ramesh, B. (2007). Agile software development: Ad hoc practices or sound principles? *IT Professional*, 9, 41–47.
10. ClickUp. (2023). ClickUp productivity platform. <https://www.clickup.com>
11. DIGITAL.AI. (2022). 16th State of Agile Report. <https://info.digital.ai/rs/981-LQX-968/images/AR-SA-2022-16th-Annual-State-Of-Agile-Report.pdf>
12. Estefan, J. A. (2008). Survey of Model-Based Systems Engineering (MBSE) Methodologies. INCOSE MBSE Initiative.
13. Friedenthal, S., Moore, A., & Steiner, R. (2015). *A Practical Guide to SysML: The Systems Modeling Language* (3rd ed.). Morgan Kaufmann.
14. Ghezzi, A., & Cavallo, A. (2020). Agile business model innovation in digital entrepreneurship: Lean startup approaches. *Journal of Business Research*, 110, 519–537.
15. Hadar, I., Sherman, S., Hadar, E., & Harrison, J. J. (2013). Less is more: Architecture documentation for agile development. In *International Workshop on Cooperative and Human Aspects of Software Engineering*.
16. Holmes, J., et al. (2022). System safety and cybersecurity within MBSE. <https://doi.org/10.4271/2024-01-3966>
17. Hooshmand, Y., Adamenko, D., Kunnen, S., & Köhler, P. (2017). An approach for holistic model-based engineering of industrial plants. In *Proceedings of the International Conference on Engineering Design*, 101–110.
18. Irnich, L., et al. (2023). Combining and evaluating MBSE solutions. <https://doi.org/10.1007/s10010-023-00619-0>
19. Kim, S. Y., Wagner, D., & Jimenez, A. (2019). Challenges in applying model-based systems engineering: Human-centered design perspective. *INCOSE Human-Systems Integration Conference*.
20. Kossiakoff, A., Biemer, S. M., Seymour, S. J., & Flanagan, D. A. (2020). *Systems Engineering Principles and Practice*. Wiley.
21. Kuchta, D. (2019). Combination of the earned value method and the Agile approach—A case study of a production system implementation. In *Intelligent Systems in Production Engineering and Maintenance* (pp. 87–96).
22. Lethbridge, T., Singer, J., & Forward, A. (2003). How software engineers use documentation: The state of the practice. *IEEE Software*, 20, 35–39.
23. Manifesto for Agile Software Development. (2001). <http://agilemanifesto.org>
24. Müller, et al. (2023). MBSE methodology and user acceptance in industry. <https://doi.org/10.1016/j.procir.2023.03.135>
25. Noguchi, R. A. (2019). A roadmap for advancing MBSE practice in government acquisition. *INCOSE International Symposium*.
26. Rinker, F., et al. (2023). Traceable multi-view model integration for agile systems engineering. <https://doi.org/10.1007/s42979-022-01572-5>
27. Ruparel, N. B. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, 35, 8–13.
28. Satpathy, T. (2016). *A Guide to the Scrum Body of Knowledge (SBOK™ Guide)* (3rd ed.). SCRUMstudy.
29. Selic, B. (2009). Agile documentation, anyone? *IEEE Software*, 26, 11–12.
30. Stettina, C. J., & Heijstek, W. (2011). Necessary and neglected? Internal documentation in agile teams. In *ACM Conference on Design of Communication*.
31. Strobbe, C., et al. (2023). Model-based multi-disciplinary design for cyber-physical systems. <https://doi.org/10.1177/10920617241295885>
32. Turner, R. (2007). Toward Agile systems engineering processes. *Crosstalk Journal of Defense Software Engineering*, 20, 11–15.
33. U.S. Government Accountability Office. (2015). *Agile Assessment Guide: Best Practices for Agile Adoption and Implementation* (GAO-20-590G).
34. Van Acker, B., et al. (2023). Validity frame-enabled model-based engineering processes. <https://doi.org/10.1177/00375497231205035>
35. Vijayarathy, L. R., & Turk, D. (2008). Agile software development: A survey of early adopters. *Journal of Information Technology Management*, XIX, 1–8.

36. Walden, D. D., Roedler, G. J., & Forsberg, K. (2015). INCOSE systems engineering handbook version 4. INCOSE International Symposium.
37. Wang, L., Izygon, M., Okon, S., Wagner, H., & Garner, L. (2016). Effort to accelerate MBSE adoption and usage at JSC. AIAA SPACE.
38. Xie, et al. (2024). X-SEM: Modeling and simulation-based systems engineering methodology. <https://doi.org/10.1016/j.jmsy.2024.01.013>
39. Yeo, K. T. (2002). Critical failure factors in information system projects. *International Journal of Project Management*, 20, 241–246.
40. Young, K. G. (2015). Defense space application of MBSE—closing the culture chasms. AIAA SPACE Conference.
41. Zimmerman, P. (2014). A review of model-based systems engineering practices and recommendations for future directions in the Department of Defense.

Financiación

Los autores no recibieron financiamiento para el desarrollo de esta investigación.

Conflictos de interés

Los autores afirman que no existen conflictos de intereses en este estudio y que se han seguido éticamente los procesos establecidos por esta revista. Además, aseguran que este trabajo no ha sido publicado parcial ni totalmente en ninguna otra revista.

Contribución de autoría

Conceptualización: KDRM
Curación de datos: KDRM
Análisis formal: KDRM
Investigación: KDRM
Metodología: KDRM
Administración del proyecto: KDRM
Recursos: KDRM
Software: KDRM
Supervisión: KDRM
Validación: KDRM
Visualización: KDRM
Redacción – Borrador original: KDRM
Redacción – Revisión y edición: KDRM