

BMM - Backlog Management Method: refinando desde la necesidad del usuario a elementos listos para el desarrollo

Vanessa Avalos¹, Leandro Antonelli^{2,3}, Julio Hurtado⁴

¹Universidad Nacional de La Plata, La Plata, Bs As, Argentina

²Lifia, Fac. de Informática, UNLP, La Plata, Bs As, Argentina

³CAETI, Facultad de Tecnología Informática, Universidad Abierta Interamericana

⁴UNICAUCA, Universidad del Cauca, Popayan, Colombia

vanessa.avalos@info.unlp.edu.ar, lanto@lifia.info.unlp.edu.ar, ahurtado@unicauca.edu.co

Abstract. Uno de los objetivos de las metodologías ágiles está puesto en la satisfacción del cliente a través de la entrega temprana y continua de software con funcionalidad de valor. El backlog es el conjunto de elementos que organiza e impulsa el desarrollo del software, a partir de una lista ordenada. Contar con un método de gestión del backlog, ayuda a que el desarrollo de software se enfoque en lo más importante y en lo que tiene mayor impacto, brinda transparencia sobre el trabajo pendiente y el progreso, fomenta la colaboración y la toma de decisiones informadas, permite realizar ajustes continuos, otorga flexibilidad, facilita la comunicación y asegura que el foco está en lo más importante para el usuario. Diversos estudios se centran en aspectos concretos del backlog, como la gestión de cambios, trazabilidad de requisitos, dependencia de requisitos y comunicación de requerimientos. Éste artículo propone un método con un enfoque sistemático y detallado para la gestión del backlog, incluye fases de definición del backlog inicial, priorización, planificación de entregas hasta lograr elementos listos para el desarrollo, con valor para el usuario. Además, este artículo describe un análisis preliminar que muestra la usabilidad y aplicabilidad del método propuesto.

Keywords. gestión del backlog, backlog, desarrollo ágil, refinamiento.

1 Introducción

“El product backlog es una lista emergente y ordenada de lo que se necesita para mejorar el producto, es la única fuente del trabajo realizado por el equipo” [13], esta lista es dinámica y versátil, y requiere de diferentes actividades para su gestión, tales como el refinamiento, la priorización, control de trazabilidad, dimensionamiento, gestión de dependencias, entre otras.

El objetivo del producto se traduce en el backlog, que describe el estado futuro esperado al finalizar una entrega. Es clave la adecuada definición, selección, priorización de las necesidades que conforman el backlog para asegurar que el software que se desarrolla aporta valor al usuario. Es necesario contar con un marco que proporcione una estructura clara y coherente para gestionar el backlog y garantizar un proceso de desarrollo eficiente y centrado en el valor del usuario. El proceso de gestión del backlog es un proceso continuo que implica aprender más sobre lo que se está desarrollando. “El backlog evoluciona a medida que evoluciona el producto y el entorno en el que se utilizará” [5].

Diferentes investigaciones se enfocan en actividades específicas relacionadas a la gestión del backlog, otras en definir un método para el desarrollo de software. Sin un método general para la gestión del backlog, es difícil lograr una estructura para gestionar el trabajo pendiente, que el equipo de desarrollo se enfoque en los elementos más importantes y de mayor impacto, la transparencia sobre el trabajo pendiente y el progreso, fomentar la colaboración, así como la toma de decisiones informadas en el desarrollo de software ágil.

Este artículo propone un método de gestión del backlog, se enfoca en lograr fases de definición desde el inicio, incluye la planificación de entregas hasta elementos listos para el desarrollo. El método propuesto muestra la complejidad de gestionar el backlog, incluye etapas desde cómo surge y su perfeccionamiento de manera continua.

El resto del artículo está organizado de la siguiente manera. La sección 2 analiza trabajos relacionados. La Sección 3 describe el método propuesto. La sección 4 proporciona evidencia sobre la aplicabilidad y usabilidad del método. Finalmente, la sección 5 presenta algunas conclusiones y trabajos futuros.

2 Trabajos relacionados

Diferentes investigaciones se enfocan en actividades específicas relacionadas a la gestión del backlog, otras se enfocan en definir un método para el desarrollo de software.

Sedano et al. [14] presentan una teoría de product backlogs, mencionan las prácticas más utilizadas en una empresa de desarrollo de software. El proceso de generar, refinar y secuenciar conceptos de solución, características, historias y finalmente, el trabajo pendiente del backlog, éste estudio dividió las observaciones en prácticas interrelacionadas y obstáculos, resume en una teoría de product backlogs, abarca el límite entre las dos vías en la metodología ágil de doble vía, la pista uno llena el backlog, con historias de usuario mientras que la pista dos vacía el backlog. Detalla cómo surge y funciona el backlog, sin embargo, no lo estructura en forma de método general para la gestión del backlog.

Wirfs-Brock et al. [17] proponen patrones para gestionar el backlog, los patrones definen actividades y responsabilidades basadas en roles. Éstos patrones son útiles para la gestión de requerimientos, sin embargo, no lo estructura como un método de gestión. Svensson et al. [16] proponen un método de evaluación del backlog, permite a las organizaciones ágiles evaluar los trabajos pendientes y garantizar que los elementos del

backlog sean suficientes para sus necesidades y estén bien alineados con el proceso de decisión, éste método se aplica una vez identificados los elementos del backlog.

En cuanto a las investigaciones enfocadas a actividades específicas, incluyen la gestión de cambios, trazabilidad de requisitos, soluciones para mejorar la comunicación y comprender los requisitos, otros cómo presentar un backlog para fundamentar las decisiones del product owner. Y muchas publicaciones se enfocan en la priorización del backlog. Además, estudios clasifican los inconvenientes en la gestión del backlog [14] y desafíos en la ingeniería de requisitos en métodos ágiles [7].

Soares [15] propone un modelo para presentar el backlog orientado a metas para fundamentar las decisiones del product owner. Cruz [4] aplica el marco Agile Upstream para ayudar al nivel ejecutivo y a los responsables de proyectos a tomar decisiones estratégicas sobre la selección, priorización y descarte de iniciativas. Sambinelli et al. [12] muestran resultados empíricos relacionados con la adopción de estrategias para aumentar valor al cliente.

Martakis et al. [8] se enfocan en las prácticas de las industrias para la gestión de dependencias de requisitos, éstas se tratan como parte de la gestión de riesgos. Elshandidy [6] propone un marco para integrar la ingeniería de requisitos basada en el comportamiento para ingeniería de línea de productos ágil.

Estas investigaciones describen cuestiones particulares de la gestión del backlog, algunas detallan experiencias en la industria. Sin embargo, no proporcionan un marco general para la gestión del backlog.

Rubin [11] y Schwaber et al. [13] se enfocan en prácticas y principios ágiles para la creación y gestión del backlog, sin embargo, no estructuran en pasos necesarios para la gestión del backlog. En lo que respecta a la planificación de entregas, abordan de manera general. El método propuesto, establece los pasos necesarios para la gestión del backlog, ofrece un enfoque más estructurado y detallado para la planificación de entregas, que abarca desde la identificación de expectativas y actores hasta la planificación y refinamiento de entregas.

La gestión del backlog y la implementación de Dosshi [5] es útil como complemento de Schwaber et al. [13], proporcionan clarificaciones y prácticas recomendadas. El método propuesto ofrece un enfoque más sistemático y detallado para la gestión del backlog, lo cual es beneficioso en contextos donde se requiere una mayor especificación y alineación con las necesidades del usuario.

3 BMM - Backlog Management Method. Método Propuesto

El método BMM propuesto se estructura en cuatro etapas, comienza con la definición del backlog inicial. En esta etapa, se busca unificar las expectativas del software mediante dinámicas colaborativas que involucran a las partes interesadas. Se establece claramente qué elementos están incluidos y cuáles no en el software, se identifican los actores y el impacto esperado y qué funcionalidades deben incluirse para satisfacer sus necesidades. Las necesidades se traducen en requerimientos a un nivel general, los cuales se estructuran en una secuencia lógica y se visualiza el flujo de trabajo del usuario para tener una comprensión clara de cómo interactuarán con el

software. Este proceso garantiza un entendimiento compartido de la experiencia del usuario y sienta las bases para las siguientes etapas de gestión del backlog.

Una vez que las necesidades generales han sido identificadas, se procede a priorizar los elementos del backlog utilizando técnicas específicas. Posteriormente, se planifican las entregas o releases del software. En la etapa de refinamiento, las épicas o historias de usuario definidas para el release, se desglosan en historias más detalladas. Una vez alcanzado un entendimiento compartido con el equipo de desarrollo, las historias de usuario refinadas están listas para ser asignadas a iteraciones de desarrollo.

El método BMM se resume en la figura 1, se estructura en fases de definición.



Figura 1. Fases de BMM - Backlog Management Method

A continuación, se describen con mayor detalle, las distintas fases de definición del método propuesto.

3.1 Definición del backlog inicial

Esta etapa logra comprender los objetivos del software, las necesidades y expectativas para determinar qué requerimientos deben incluirse en el backlog inicial.

Con el objetivo de unificar las expectativas del software, se realizan diferentes dinámicas con todas las partes involucradas para consensuar el objetivo del software, se determinan aquellos requerimientos de alto nivel, que serán foco y aquellos que NO estarán en el producto. Una vez definido el objetivo e identificado lo que se incluirá en el software, se identifican las personas o roles que intervienen en el software.

Esta etapa logra que todos los involucrados construyan una única definición del objetivo del producto, sepan porqué se construye, qué no está incluido, conozcan el mapa de los roles que participan directamente y los roles que van a interactuar eventualmente.

La Figura 2, muestra un ejemplo de un mapa de roles del software generado en uno de los proyectos donde se aplicó el método.



Figura 2. Mapa de roles del software

Luego se identifica qué impacto se espera para cada actor, es decir, cómo se quiere cambiar la vida de éstos actores, según los objetivos del software. A través de la técnica de Impact Mapping, se crea un mapa mental.

De manera colaborativa se obtiene un listado de las necesidades a nivel general para cada actor involucrado. La Figura 3 muestra un fragmento del impacto esperado en uno de los proyectos.



Figura 3. Mapa de impacto

A partir de las necesidades identificadas en el paso anterior, en ésta etapa, se organizan y estructuran en una secuencia que describe los pasos, permite visualizar el flujo de trabajo del usuario y las características a implementar. Se desglosan las necesidades en épicas o historias de usuario gestionables. A través de la técnica de User Story Mapping. Tal como lo sugiere Patton [9] “Concéntrese en la amplitud de la historia antes de profundizar en ella”. En esta etapa aún se tratan las necesidades esperadas a nivel general.

3.2 Priorización del backlog

En base al mapa de historias de usuario, se procede a realizar la valoración de las historias identificadas, según la técnica MoSCoW u otra técnica por ejemplo talla de camiseta.

En esta etapa es clave asegurar que los elementos más importantes se abordan primero, se tienen en cuenta criterios como el valor que aporta al usuario, la complejidad, la dependencia de otros elementos, el riesgo, la fecha límite, entre otros. Se asignan prioridades a los elementos del backlog y luego se ordenan según el valor de prioridad asignado. Esta etapa logra un backlog priorizado, con los elementos

claramente identificados, los más críticos o de mayor valor son los que tienen prioridades más altas.

En los proyectos donde se aplicó este método, durante la etapa de priorización, los stakeholders demostraron un entendimiento sólido y claro de los elementos del backlog, lo que permitió una priorización fluida.

3.3 Planificación de entregas

Una vez priorizados los elementos del backlog, se inicia la planificación de qué elementos forman parte de las entregas o releases del software en función de sus prioridades. Esto permite que el equipo de desarrollo se enfoque en lo más importante para el usuario. Se sigue la premisa de lograr entregas continuas y pequeñas.

La Figura 4 muestra un fragmento del plan de entregas con releases definidos, luego de la planificación.

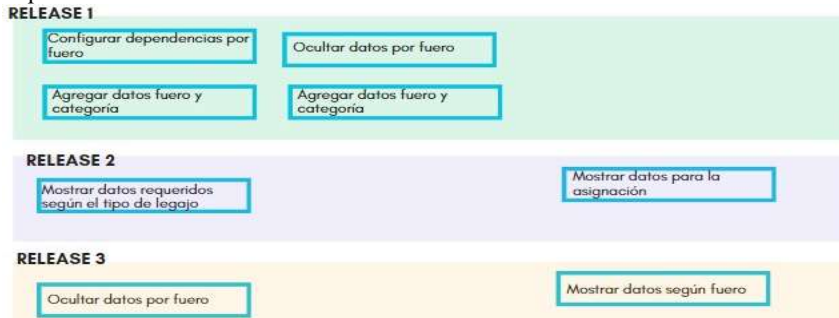


Figura 4. Parte del backlog con planificación de entregas

3.4 Refinamiento

Si bien durante las etapas previas se fue realizando el proceso de refinamiento, en esta etapa se realizan tareas de refinamiento en las historias de usuario que corresponden al release definido, estas tareas están destinadas a mejorar la calidad y claridad de las historias de usuario. Participa el equipo de desarrollo, referentes y usuarios que sean clave en la definición de los detalles de las historias. En esta etapa se considera lo expresado por Pokharel et al. [10] “las historias de usuarios deben escribirse de tal manera que el valor para el cliente sea claro”.

Se realizan conversaciones que permiten la comprensión del equipo de desarrollo, se ajustan detalles en la redacción de las historias, se descomponen las historias en tareas más pequeñas y manejables para ser implementadas por el equipo de desarrollo. Esta etapa se cierra con historias de usuario bien definidas, priorizadas y listas para ser trabajadas, minimizando las ambigüedades, es decir, comprendidas por el equipo de desarrollo.

Aunque estas fases se describen de manera separada, es posible que en cualquiera de ellas se identifiquen nuevas iniciativas, características o se desglosen historias de

usuario, esta situación requiere pasar nuevamente por fases previas, si es necesario. La figura 5 representa las entradas y salidas de cada fase.

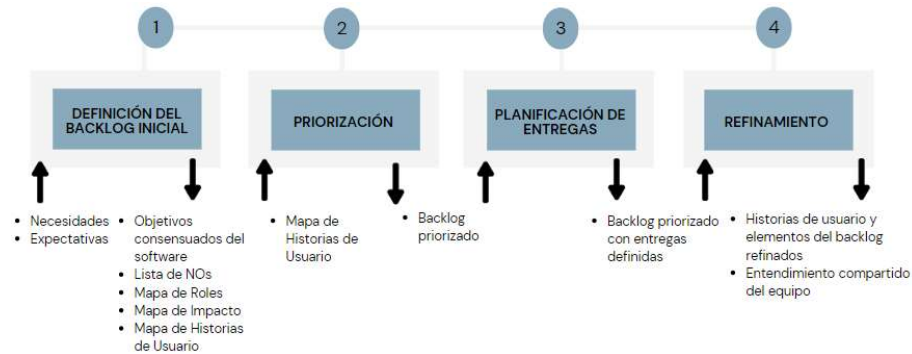


Figura 5. Entradas y salidas en cada fase de método BMM

La repetición del ciclo de gestión del backlog está relacionada con el proceso iterativo y continuo de planificación, priorización para la entrega de valor al usuario. Después de cada iteración se debe realizar una revisión y ajuste continuo de las prioridades basado en el feedback recibido y los cambios en las condiciones del software, es decir, el equipo puede tener que enfocarse en nuevas iniciativas o en mejorar las características existentes en lugar de seguir agregando elementos al backlog.

4 Validación

El método BMM se aplicó en tres proyectos de desarrollo de software, uno de ellos de tipo “greenfield” y 2 de tipo “brownfield”, todos relacionados al dominio de la gestión de justicia, con un plazo establecido de tres y cuatro años de entrega del producto completo y con entregas graduales. Dado el alcance de éstos proyectos, se requiere de una asignación esencial en términos de tiempo y recursos para gestionar adecuadamente el backlog.

Estos proyectos son parte del reciente proceso de transformación en la justicia, que incluye reformas con impacto sustancial en la forma de impartir justicia, implicó la redefinición de procesos, nuevas medidas y reglamentos y en la necesidad de contar con proyectos de desarrollo de software que se ajusten a los nuevos paradigmas jurídicos. La naturaleza jurídica de estos proyectos puede implicar backlogs relativamente estables, sin embargo, el tamaño, la complejidad, cantidad de usuarios y el impacto significativo de los proyectos generan desafíos en la gestión y priorización de requerimientos de manera organizada, para asegurar un enfoque definido. Otro factor determinante es el reciente cambio de paradigma jurídico, que exige que los referentes y stakeholders comuniquen sus necesidades de software, cuando aún cuentan con pocos años de experiencia en su forma de impartir justicia con el nuevo paradigma. Por ésta razón, se optó por la metodología de desarrollo de desarrollo ágil, que permite

gestionar requerimientos sujetos a cambios a medida que se obtiene feedback de los usuarios.

En la validación del método BMM participaron 20 profesionales que forman parte de los proyectos, con diferentes roles, gerentes del área de informática, project manager, analistas, desarrolladores, QC-testers, arquitectos de datos, devops. Estos participantes tienen más de un año de experiencia en la aplicación del método propuesto.

Se proporcionó la información y gráfica del método a los participantes, para lograr una adecuada evaluación en la aplicabilidad del método. Se utilizó la herramienta Formularios de Google, para distribuir el cuestionario, el cual fue de carácter anónimo.

Se utilizó la escala de usabilidad del sistema (SUS) [2] [3] para evaluar la usabilidad y aplicabilidad del método. Aunque SUS se utiliza principalmente para evaluar la usabilidad de sistemas de software, se ha demostrado que es eficaz para evaluar productos y procesos [1].

La escala de Usabilidad del Sistema (SUS) es una escala simple de diez ítems que brinda una visión global de las evaluaciones subjetivas de usabilidad. SUS es una escala Likert, las preguntas del cuestionario cubren una variedad de aspectos de la usabilidad, como la necesidad de soporte, capacitación y complejidad y, por lo tanto, tienen un alto nivel de validez aparente para medir la usabilidad de un sistema. [2] [3]

Cada pregunta debe responderse en una escala de cinco opciones, que van desde "1" "Totalmente en desacuerdo" hasta "5" "Totalmente de acuerdo". Las preguntas se relacionan por parejas, planteando la misma pregunta, pero desde un punto de vista complementario para obtener un resultado de alta confianza.

Para el cálculo de la puntuación SUS los ítems 1, 3, 5, 7 y 9 se califican considerando el valor clasificado menos 1. Luego, los ítems 2, 4, 6, 8 y 10 se califican considerando 5 menos el valor clasificado. Después de eso, las puntuaciones de cada participante se suman y luego se multiplican por 2,5 para obtener un nuevo valor que oscila entre 0 y 100. Finalmente, se calcula el promedio. Los posibles resultados son: "No aceptable" 0-64, "Aceptable" 65-84 y "Excelente" 85-100. La puntuación obtenida fue 71,87, por lo tanto, el método BMM puede considerarse aceptable.

5 Conclusiones

Wirfs-Brock et al. [17], afirman que “Para sistemas y equipos grandes, el trabajo pendiente no puede ser una lista de tareas pendientes no estructurada, sino que debe diseñarse e implementarse para respaldar el proceso de desarrollo del equipo. El backlog es más que una simple lista de tareas pendientes para el equipo. Capta el contrato con las partes interesadas a través del detalle de los requisitos, respalda la planificación del equipo y proporciona una gran cantidad de información dependiendo de cómo se utilizan y mantienen los atributos de los diversos elementos del backlog”.

Diferentes investigaciones presentan una teoría del backlog, un método de evaluación del backlog, patrones para gestionar el backlog, otros estudios describen actividades específicas relacionadas a la gestión del backlog: la trazabilidad de requisitos, soluciones para mejorar la comunicación y comprender los requisitos, la gestión de cambios, cómo presentar un backlog para fundamentar las decisiones del

product owner. Sin embargo, hacen falta enfoques sistemáticos para la gestión del backlog.

Es necesario contar con un método general para la gestión del backlog que brinde una estructura para gestionar el trabajo pendiente, que permita al equipo de desarrollo se enfoque en las tareas de mayor valor para el usuario, que facilite la transparencia sobre el trabajo pendiente, que fomente la colaboración y la toma de decisiones informadas en el desarrollo de software ágil.

Nuestro artículo propone el método de gestión BMM - Backlog Management Method, es un marco definido para lograr resolver las dificultades anteriormente mencionadas. El método se enfoca en lograr etapas de definición desde el inicio hasta elementos listos para ser asignados a iteraciones de desarrollo. Comienza con la definición del backlog inicial, donde se unifican las expectativas y se identifica el impacto esperado mediante un mapa mental colaborativo. Se identifican las iniciativas y se avanza hacia un proceso de comprensión compartida de la experiencia del usuario, siguiendo un flujo de trabajo definido. Después de la identificación del backlog inicial, se llevan a cabo actividades de priorización para garantizar que los elementos más importantes se aborden primero. A continuación, se planifican las entregas. Esta etapa es crucial para mantener el enfoque en lo más prioritario. Posteriormente, se realiza el refinamiento final de las historias de usuario para el release definido.

El método BMM fue evaluado por 20 profesionales, con diferentes roles con más de un año de experiencia en la aplicación del método BMM. Para la evaluación se utilizó la escala de usabilidad del sistema (SUS) [2] [3] para evaluar la usabilidad y aplicabilidad del método, a través de un cuestionario de 10 ítems con opciones de respuesta en una escala de 1 a 5, los profesionales indicaron su nivel de acuerdo o desacuerdo. La puntuación final por encima de 68 se considera buena usabilidad, mientras que puntuaciones inferiores indican áreas que requieren mejora. La puntuación obtenida fue 71,87, por lo tanto, el método puede considerarse aceptable.

El enfoque propuesto surge de los años de experiencia en el desarrollo de software ágil, la gestión de proyectos grandes y de la necesidad de estructurar en pasos para lograr una efectiva gestión del backlog. Se realizarán trabajos adicionales para obtener evaluaciones más completas y exhaustivas, que permitan mejorar la validez del método, se podrán incluir evaluaciones de aplicabilidad en situaciones particulares y en otros proyectos de desarrollo.

6 Referencias

1. Bangor, A., Kortum, P. T. and Miller, J. T. "An empirical evaluation of the system usability scale" *International Journal of Human-Computer Interaction* 24.6, pp. 574-594. DOI: 10.1080/10447310802205776. 2008.
2. Brooke, J. "SUS: a retrospective." *Journal of usability studies* 8.2, pp.29-40, 2013.
3. Brooke, J. "SUS-A quick and dirty usability scale. *Usability evaluation in industry*", 189 (194), 4-7, 1996.
4. Cruz, F. "Agile upstream framework: Uma forma ágil de selecionar, priorizar e descartar iniciativas como apoio a donos de produto, executivos e escritórios de

projetos” MundoPM - Out/Nov Sumário – Edição 95 ISSN 1807-8095 pp 8-47. 2020.

5. Dosshi, H. Scrum insights for practitioners, The Scrum guide companion, editorial practiceagiles.com. 2016.
6. Elshandidy, H. "Behaviour-Driven requirements engineering for agile product line engineering" IEEE 27th International Requirements Engineering Conference (RE), Jeju, Korea (South), pp. 434-439, doi: 10.1109/RE.2019.00058. 2019.
7. Gupta A., Poels G. and Bera, P. "Using conceptual models in agile software development: A possible solution to requirements engineering challenges in agile projects," in IEEE Access, vol. 10, pp. 119745-119766, doi: 10.1109/ACCESS.2022.3221428. 2022.
8. Martakis, A. and Daneva, M. "Handling requirements dependencies in agile projects: A focus group with agile software development practitioners" IEEE 7th International Conference on Research Challenges in Information Science (RCIS), Paris, France, pp. 1-11, doi: 10.1109/RCIS.2013.6577679. 2013.
9. Patton, J. User story mapping, discover the whole story, build the right product, editorial O'Reilly Media. 2014.
10. Pokharel, P. and Vaidya, P. "A study of user story in practice" International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), Sakheer, Bahrain, pp. 1-5, doi: 10.1109/ICDABI51230.2020.9325670. 2020.
11. Rubin K. Essential Scrum, a practical guide to the most popular agile process, editorial Addison-Wesley. 2013.
12. Sabinelli, F. and Borges, M. A. F. "Survey on strategies to increase customer value in Brazilian agile software development companies" 14th Iberian Conference on Information Systems and Technologies (CISTI), Coimbra, Portugal, pp. 1-7, doi: 10.23919/CISTI.2019.8760653. 2019.
13. Schwaber, K. and Sutherland, J. "La guía de scrum". <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>. Accedido el 15 de diciembre de 2023.
14. Sedano, T., Ralph, P. and Péraire, C. "The product backlog" IEEE/ACM 41st International Conference on Software Engineering (ICSE), Montreal, QC, Canada, pp. 200-211, doi: 10.1109/ICSE.2019.00036. 2019.
15. Soares, R.M. "Product backlog orientado a metas em projetos scrum para fundamentar as tomadas de decisões do product owner" 91f. Dissertação (Mestrado em Sistemas e Computação) - Centro de Ciências Exatas e da Terra, Universidade Federal do Rio Grande do Norte, Natal, 2020.
16. Svensson, R.B., Gorschek, T., Bengtsson, P.O., Widerberg, J. "BAM - Backlog assessment method". In: Kruchten, P., Fraser, S., Coallier, F. (eds) Agile Processes in Software Engineering and Extreme Programming. XP. Lecture Notes in Business Information Processing, vol 355. Springer, Cham. https://doi.org/10.1007/978-3-030-19034-7_4. 2019.
17. Wirfs-Brock, R., Hvatum, L. "Even more patterns for the magic backlog" PLoP '18: Proceedings of the 25th Conference on Pattern Languages of Programs. October Article No. 17 Pages 1–17. 2018.