
IASB[®] meeting

Date	October 2026
Project	Intangible Assets
Topic	Recognition of an intangible asset and supporting requirements (agile software development test case)
Contacts	Hyunseon Hong (hyunseon.hong@ifrs.org)
	Vanessa Hanna (vanessa.hanna@ifrs.org)
	Jelena Voilo (jvoilo@ifrs.org)
	Tim Craig (tcraig@ifrs.org)

This paper has been prepared for discussion at a public meeting of the International Accounting Standards Board (IASB). This paper does not represent the views of the IASB or any individual IASB member. Any comments in the paper do not purport to set out what would be an acceptable or unacceptable application of IFRS[®] Accounting Standards. The IASB's technical decisions are made in public and are reported in the IASB[®] *Update*.

Purpose of the paper

1. One of the initial work streams of the Intangible Assets project is to explore whether to update some aspects of requirements in IAS 38 *Intangible Assets* relating to the definition of an intangible asset and the recognition of intangible assets to make those requirements easier to apply, particularly to newer types of intangible assets. In this paper, we explore potential approaches to addressing some recognition issues identified by stakeholders when applying IAS 38 to newer types of intangible assets—based on agile software development, one of the selected test cases.
2. This paper explores challenges relating to the unit of account, the distinction between the research and development phases and the application of the recognition criteria for internally generated intangible assets.
3. This paper asks members of the International Accounting Standards Board (IASB) for their comments and questions on our analysis, the approaches considered for improving recognition requirements for intangible assets and our next steps. It does not ask the IASB to make any decisions.

Structure of the paper

4. This paper includes:
- (a) background information (paragraphs 5–9);
 - (b) our approach to exploring potential improvements to recognition requirements (paragraphs 10–14);
 - (c) our considerations of potential improvements to IAS 38 to respond to identified challenges associated with:
 - (i) the unit of account (paragraphs 15–38);
 - (ii) the two-phase model that distinguishes between research and development (paragraphs 39–59); and
 - (iii) the six additional recognition criteria for internally generated intangible assets (paragraphs 60–66);
 - (d) next steps for exploring the agile software development test case (paragraph 67–69);
 - (e) questions for IASB members (following paragraph 69);
 - (f) Appendix A—Summary of FASB Accounting Standards Update 2025-06; and
 - (g) Appendix B—Fact pattern explored in this paper.

Background information

Characteristics of agile software development

5. Under a traditional (waterfall) product development methodology, development processes are formalised through linear and sequential phases, including identification, design, development, testing, deployment and maintenance. Each phase must be substantially completed before the subsequent phase can commence, creating clear stage gates and documentation requirements. For example, in the pharmaceutical industry, entities are required to follow predetermined sequential processes to develop

new medicines in accordance with jurisdictional regulations, where rigorous documentation and phased clinical testing completion are essential for regulatory approval and patient safety.

6. In this traditional development environment, products are delivered only upon completion of all procedural steps, which can span months or even years depending on project complexity. Consequently, customer or internal user preferences cannot be readily incorporated beyond the initial requirement gathering and final user-acceptance phases due to the relatively rigid and stable characteristics of the process.
7. To respond to rapidly changing technologies, shortened product lifecycles and the need to meet customer preferences in a timely manner, entities have increasingly adopted agile development methodologies, particularly in the software industry. Agile software development is intended to enable entities to deliver software to customers or internal users more efficiently. Compared to traditional software development, the key characteristics of agile software development include:
 - (a) iterative cycles: software is developed and enhanced through repeated cycles (iterations), rather than being delivered comprehensively at the end of an extended or milestone-driven development cycle. In many agile development methodologies, work is organised into short, time-based iterations, such as sprints. The development might often fail quickly, but an entity learns from the failure and moves on to the next iteration.
 - (b) accelerated delivery: the process facilitates accelerated delivery of software that aligns with customer or internal user requirements. Working software is delivered in small increments over short time cycles, enabling continuous feedback and iterative improvements.
 - (c) an adaptable process: the approach is highly adaptable and responsive to change. It accommodates evolving customer or internal user requirements, even during later stages of development, to ensure the software product remains relevant and valuable to users.

Recap of the IASB's previous discussion

8. In [January 2026](#), the IASB discussed stakeholder concerns about applying IAS 38 to agile software development. Those concerns relate to how agile software development methodologies affect the recognition of development costs, in particular the timing of recognition and the challenges associated with identifying costs related to research, development and maintaining the intangible asset after initial implementation. Stakeholders commonly highlighted challenges relating to:
- (a) determining the unit of account;
 - (b) distinguishing between research and development phases;
 - (c) applying the six recognition criteria for capitalising development costs, particularly the criteria relating to technical feasibility and probable future economic benefits; and
 - (d) applying the initial and subsequent measurement requirements, including when to begin amortisation (see paragraph 69).
9. In response to the concerns, the IASB discussed exploring whether, by applying the updated recognition criteria and supporting concepts in the *Conceptual Framework for Financial Reporting (Conceptual Framework)*, it could improve the requirements in IAS 38 by providing a better basis for preparers' judgements, resulting in more useful information, about intangible assets developed using different development methods. IASB members also discussed the possibility of drawing on the recent amendments to US GAAP requirements regarding research and development costs in software development (see Appendix A).

Approach to exploring potential improvements to recognition requirements

10. This paper first explores approaches to determining the unit of account before considering related questions regarding the distinction between research and development phases and the application of the recognition criteria.

-
11. We believe this is an appropriate starting point because:
- (a) stakeholders identified determining the unit of account as one of the most challenging judgements in applying the recognition requirements in IAS 38, particularly in agile software development environments characterised by iterative development and frequent enhancements.
 - (b) determining the unit of account logically precedes the application of the recognition requirements, and different conclusions about the unit of account can affect the assessment of the recognition criteria, the timing and amount of capitalisation and subsequent accounting outcomes. Therefore, considering the unit of account provides a more meaningful foundation for the subsequent analysis.
12. For each topic discussed in this paper, we explore possible approaches and alternatives that could address the challenges raised by stakeholders, consider their implications and provide initial staff leanings.
13. Although stakeholders also identified challenges relating to initial and subsequent measurement, this paper focuses on challenges related to recognition. We think it is necessary to make sufficient progress on the unit of account and recognition requirements first because conclusions on those topics may affect the analysis of measurement and amortisation. As discussed in paragraph 69, we plan to consider certain aspects of the measurement requirements, including when to begin amortisation, in a future meeting.
14. We do not suggest that the alternatives explored in this paper represent the only possible approaches nor do we suggest that any of the alternatives explored should be adopted, even those that may align more closely with the staff's initial leaning. Rather, these alternatives are intended to provide a foundation for further analysis, discussion and stakeholder consultation.

Unit of account

Stakeholder challenges

15. As discussed in paragraph 11(a), stakeholders expressed concerns about determining the appropriate unit of account when applying IAS 38 to agile software development. These concerns arise because, unlike traditional development methods, agile software development is characterised by iterative cycles and accelerated delivery, whereby software is developed and enhanced through repeated iterations, such as sprints, and delivered in small increments. As a result, each iteration may result in a piece of software that provides some functionality. Stakeholders said this can make it challenging to determine whether pieces of software developed through those iterations should be considered separately or together, and consequently, at what level the unit of account should be identified. This challenge is made more difficult by the adaptable process characteristic of agile software development, which allows functionality and requirements to evolve throughout the development process.

Our responses

Factors to be considered

16. As a starting point, and consistent with the approach applied in the [September 2026](#) software as a service (SaaS) test case paper, we considered the *Conceptual Framework* and IFRS Accounting Standards in exploring how the unit of account might be determined for software, and potentially other intangible assets, developed using agile methodologies.
17. In considering how the unit of account might be determined, we think that the principles in paragraphs 4.48–4.55 of the *Conceptual Framework* could be helpful. Paragraph 4.48 of the *Conceptual Framework* defines the unit of account as the right, or group of rights, to which recognition criteria and measurement concepts are

applied. Paragraph 4.51 specifies that a unit of account is selected to provide useful information, which implies that:

- (a) the information must be relevant. For example, one indicator in paragraph 4.51(a) that could be considered is whether treating a group of rights as a single unit of account provides more relevant information than treating each right as a separate unit of account if those rights are used together in the business activities conducted by an entity to produce cash flows and are measured by reference to estimates of their interdependent future cash flows.
- (b) the information must faithfully represent the substance of the transaction. For example, faithfully representing the substance of the transaction may require an entity to treat rights arising from pieces of software developed as part of a project as a single unit of account, or to separate those rights into separate units of account.

- 18. Paragraphs 22–30 of IFRS 15 *Revenue from Contracts with Customers* provide a framework for identifying performance obligations by assessing whether promised goods or services are distinct. In particular, paragraphs 27–29 of IFRS 15 consider whether a good or service is capable of being distinct and whether it is distinct within the context of the contract.
- 19. Although IFRS 15 applies in the context of contracts with customers and internally generated intangible assets are not necessarily linked to contractual arrangements, we think some of the considerations in paragraphs 27–29 of IFRS 15 could be helpful in exploring how the unit of account might be determined for internally generated software. In particular, an entity could consider:
 - (a) whether it can obtain benefits from exercising its rights over a piece of software on its own or together with other resources readily available to the entity.
 - (b) whether the right over a piece of software is separately identifiable from rights over other pieces of software the entity is developing. In making that assessment, an entity could consider:

-
- (i) whether pieces of software are inputs to a combined output;
 - (ii) whether pieces of software significantly modify one another; and
 - (iii) whether pieces of software are highly interdependent or highly interrelated.
20. These considerations may help an entity assess whether rights arising from separate pieces of software should be accounted for separately or together. In IFRS 15, the customer contract provides the basis for assessing whether promised goods or services should be considered separately or together. In the absence of an equivalent contractual arrangement for internally generated software, an entity may also consider management's intended use of the software when assessing whether rights arising from separate pieces of software should be considered separately or as part of a broader grouping.

Underlying item

21. As discussed in paragraphs 16–20, in determining the appropriate unit of account, an entity would need to identify the rights it obtains when developing software (or another intangible asset) and consider whether those rights are separately identifiable and how economic benefits are expected to be generated from those rights. To identify those rights, the entity would need to consider the underlying items over which they are held.
22. In the [July 2026](#) SaaS test case paper, the underlying item was identified as the intellectual property (IP). We think that the underlying item in agile software development is also the IP. However, unlike in a SaaS arrangement, where the underlying item is IP in the form of developed software, agile software development creates multiple pieces of IP each with some functionality throughout the development process.
23. An entity developing IP controls rights over that IP. Those rights are rights to direct the use of the IP (see Figure 1 in the [September 2026](#) SaaS test case paper). Through

those rights, an entity may generate economic benefits by selling or licensing the IP, combining it with other economic resources to produce goods or provide services, or using it to enhance the value or performance of other economic resources.

24. In agile software development, individual time-based iterations, such as sprints, create pieces of IP with some functionality. Consequently, the rights the entity controls are rights to direct the use of those multiple pieces of IP, each of which has some functionality.
25. Determining the appropriate unit of account therefore involves identifying the appropriate grouping of rights to direct the use of the multiple pieces of IP from which the entity expects to generate economic benefits.

Functionality

26. Having identified the underlying items and the rights over those items, an entity would then need to consider whether those rights are separately identifiable and how it expects to generate economic benefits from those rights. We think this depends on the functionality of the underlying IP and management's intended use of that IP.
27. Management's assessment of how economic benefits will be generated may differ depending on whether the software is intended for sale or internal use. For example, an entity developing software for sale may consider whether sufficient functionality has been developed for the software to be made available to customers, whereas an entity developing software for internal use may consider whether sufficient functionality has been developed for the software to be used in its operations. However, we think that in both cases, the objective is the same because the entity is assessing whether the developed IP functionality is sufficient for the entity to generate economic benefits from its rights over the pieces of IP and, consequently, the appropriate grouping of those rights.
28. Although individual time-based iterations, such as sprints, create pieces of IP with some functionality, we think that a grouping based on the output of a single iteration

is likely to be too narrow when determining the appropriate unit of account.

Management is typically not assessing how economic benefits are expected to be generated on the basis of the output of a single iteration. Instead, management considers IP on a broader—typically project—level when assessing how it expects to generate economic benefits, which may in turn help identify the appropriate grouping of rights over the related pieces of IP for the purposes of determining the unit of account.

29. Accordingly, we think that the determination of the appropriate unit of account should be considered at a broader project level rather than at the level of an individual iteration.¹ The following section explores alternative approaches to determining the unit of account at a broader project level.

Exploring options for the unit of account

30. The analysis in paragraphs 26–29 suggests that linking the unit of account to a project-level grouping may be more appropriate than linking it to an individual iteration. However, given the typical complexity of modern software development projects, identifying the project-level grouping may not, on its own, determine the appropriate unit of account. Different approaches may be taken for determining the appropriate aggregation of pieces of IP with some functionality developed as part of a project for identifying the relevant project-level grouping and, consequently, the appropriate unit of account.
31. To explore these approaches, we developed a fact pattern based on software developed using an agile methodology. Using that fact pattern, we considered three alternative views on determining the appropriate aggregation of IP functionality.
32. The following section is structured as follows:
- (a) paragraph 33 provides a summary of the three views;

¹ US GAAP identifies a 'software project' as the unit of account but does not define the term. Appendix A provides further information on the Financial Accounting Standards Board's (FASB's) decision not to define what constitutes a software project

-
- (b) paragraphs 34–38 set out our observations, including an initial staff leaning; and
 - (c) Appendix B sets out the underlying fact pattern and detailed analysis of each view.
33. The views considered are:
- (a) View A: the unit of account is linked to all functionality approved by management as part of a single project. The recognition criteria are therefore assessed at the level of the overall project. Of the approaches considered, this approach would typically result in the largest unit of account.
 - (b) View B: the unit of account is linked to distinct functionality. The recognition criteria are assessed separately for each feature or grouping of features that provides distinct functionality. Of the approaches considered, this approach would typically result in the smallest units of account.
 - (c) View C: the unit of account is linked to the smallest aggregation of functionality from which management expects to generate future economic benefits. The recognition criteria are therefore assessed at the level of that aggregation. Of the approaches considered, this approach would typically result in units of account that fall between View A and View B.

Initial staff leaning

34. The staff's initial leaning is that View C may provide the most appropriate approach to identifying a unit of account. View C links the unit of account to the smallest aggregation of functionality from which management expects to generate future economic benefits. In the fact pattern, features 1–3 represent that aggregation because together they comprise the minimum functionality required for an initial release and the first meaningful differentiation from previous releases.
35. We think View C may provide the most appropriate approach because:

-
- (a) features 1–3 represent the point at which management expects to generate future economic benefits and together the features are expected to contribute to the cash flows generated by the game. Grouping the rights arising from those features as a single unit of account may therefore provide more relevant information.
- (b) features 1–3 are intended to be used and released together to provide the combined output necessary for the initial release. Grouping the rights arising from those features as a single unit of account may therefore provide a more faithful representation than treating those rights separately.
36. By contrast, we observe that View A and View B may identify the unit of account at levels that do not align with the way management expects to generate future economic benefits. Specifically:
- (a) View A may be simpler to apply because all functionality approved as a single project is assessed together, largely consistent with how projects are typically approved and funded. However, it may result in a unit of account that is too broad by linking it to an aggregation of functionality that includes functionality with low technical uncertainty and functionality that remains subject to significant technical uncertainty. As a result, it may delay recognition and provide less relevant information to users of financial statements.
- (b) View B may allow recognition assessments to be performed as uncertainty is resolved for individual features. However, by focusing on distinct functionality, it may result in a unit of account that is too narrow. In the fact pattern, features 1–3 are intended to operate together as an integrated game experience and provide the combined output necessary for the initial release. Recognising separate assets linked to specific features may therefore not faithfully represent the underlying economics because management does not intend to release or expect to generate future economic benefits from those features separately.

-
37. We believe that View C could also provide a practical boundary for identifying a project-level unit of account. In the fact pattern, once features 1–3 have been identified as the appropriate aggregation of functionality, features 4 and 5 would generally be assessed separately rather than being included within that unit of account. This would result in amortisation of the asset comprising features 1–3 beginning when those features are released on 15 October 20X1, rather than being deferred while additional functionality continues to be planned or developed. In this way, View C acts as a guardrail against continually broadening the boundary of the unit of account to include additional planned functionality, which could delay recognition and subsequent amortisation in View A.
38. The above analysis, including the staff's initial leaning towards View C, is intended to explore possible approaches to determining the unit of account. How any such approach might ultimately be reflected in IAS 38 would require further consideration. For example, the IASB could explore whether to introduce specific requirements, provide application guidance or establish a rebuttable presumption. The IASB could also consider providing indicators to assist entities in identifying the appropriate unit of account, especially in relation to the underlying IP functionality and management's intentions.

Two-phase model that distinguishes between research and development

Stakeholder challenges

39. According to IAS 38, an intangible asset is recognised only if it is probable that the expected future economic benefits that are attributable to the asset will flow to the entity and the cost of the asset can be measured reliably (paragraph 21 of IAS 38). For internally generated intangible assets, IAS 38 sets out additional requirements for distinguishing between the research and development phases including (paragraphs 52–59 of IAS 38):

-
- (a) an entity is required to distinguish the generation of an internally generated intangible asset into a research phase and a development phase. Expenditure on research, or on the research phase, is recognised as an expense when incurred, whereas expenditure incurred in the development phase is recognised as an intangible asset only if six specified criteria, including the technical feasibility, are met.
- (b) if an entity cannot distinguish the research phase from the development phase, the entity treats the expenditure on that project as if it were incurred in the research phase only.
40. Agile software development is characterised by an iterative and incremental approach in which software is progressively developed through short, time-based iterations. The results of each cycle may lead an entity to refine requirements, reconsider design decisions or explore alternative solutions in subsequent cycles. Consequently, activities that could relate to research or development phases may arise or recur within each iteration as the software evolves, rather than necessarily occurring in a clear sequential order.
41. Many stakeholders said that agile development methods lack specific and clear research and development phases.

Our responses

Economic substance of agile and traditional (waterfall) development methods

42. Before considering issues arising from the two-phase approach and exploring potential improvements to recognition requirements, we think it is important to understand the underlying economic substance of agile and waterfall development methodologies. If agile development is fundamentally different in economic substance from traditional (waterfall) development, IAS 38, which was developed in the context of more traditional development approaches, might require more fundamental

changes, including the development of new recognition requirements for intangible assets developed using agile methodologies.

43. Although the agile and traditional (waterfall) development methodologies differ in how development activities are organised and executed, we think their underlying economic substance is fundamentally similar because:
- (a) both methodologies are undertaken with the same underlying objective of developing a specific asset or achieving a defined outcome. While agile development may sometimes begin with a broader vision and evolve over time, this does not mean that development activities lack direction or intended outcomes. In both methodologies, entities make decisions under uncertainty, commit economic resources and seek to progress development through ongoing learning and feedback.
 - (b) the principal difference lies in how the development process responds to uncertainty. Traditional (waterfall) methodologies generally seek to address uncertainty through detailed upfront planning and sequential development stages. By contrast, agile development methodologies use iterative development and shorter planning cycles, allowing an entity to test and assess incremental outputs, obtain feedback and adjust its requirements, design or develop approaches as the development progresses. This iterative process of agile development facilitates earlier identification of approaches that may not work as intended and enables the entity to learn from those outcomes and adapt.
44. From a recognition perspective, a key judgement in both methodologies relates to the likelihood of successful development. Accordingly, the fundamental accounting challenge remains the same under either approach—determining when, and on the basis of what evidence, an entity can demonstrate that the criteria for recognising the asset have been met.

Potential approaches to updating the recognition requirements

45. If the underlying economic substance of agile and traditional development methodologies is not significantly different, we think that it would be inappropriate to have separate recognition requirements for agile development methodologies. Instead, two broad approaches to updating the recognition requirements could be considered, differing by whether potential alternatives are explored within the framework of the existing recognition requirements in IAS 38:
- (a) Option 1: using or adapting the existing IAS 38 requirements for agile development—this approach would seek to develop potential solutions within the framework of the existing recognition requirements. For example, it could involve clarifying how the existing recognition requirements in IAS 38 should be applied to agile development methodology, without fundamentally changing those requirements.
 - (b) Option 2: developing a new comprehensive recognition model (fresh-start approach)—this approach would involve developing a single set of recognition requirements that would apply consistently to all development methodologies, without relying on the existing recognition requirements in IAS 38 as the starting point.

Initial staff leaning

46. We believe that starting with the existing requirements (Option 1) would be an appropriate and practical first step because:
- (a) developing a completely new recognition model on a fresh-start basis could introduce significant uncertainty and be unnecessarily disruptive. In addition, a fresh-start approach may require substantial time and resources, which could delay any improvement to the recognition requirements.
 - (b) stakeholders have provided limited calls for a fundamental change of existing recognition requirements.

-
- (c) in some industries, the existing requirements are well established and supported by extensive application experience and a well-developed body of practice.
47. If, through our exploration to find potential approaches, we identify areas in which the existing requirements do not produce appropriate outcomes, particularly for intangible assets developed through an iterative development process, we could consider targeted amendments to those requirements.
48. Although we believe that building on the existing requirements is the appropriate starting point, this would not rule out a fresh-start approach at a later stage. If the analysis indicates that more fundamental changes are needed, the IASB could then consider developing a new recognition model. In doing so, the IASB could explore a range of possible approaches, including, for example, drawing on the Financial Accounting Standards Board's (FASB's) previous work on recognition requirements for internal-use software where appropriate. In that work, the FASB removed all references to software development project stages so that the recognition requirements are neutral to different software development methods. The amendments require an entity to consider whether there is significant uncertainty associated with the development activities of the software in evaluating the recognition threshold (see Appendix A).

Potential approaches to adapting the current two-phase model

49. In considering how to adapt the existing two-phase model, we have identified three possible approaches:
- (a) Option A: focused on identifying research and development activities;
 - (b) Option B: retaining the two-phase model with additional guidance; and
 - (c) Option C: removing the requirement to distinguish between the research and development phases.

Option A

50. This approach focuses on the nature of the activity giving rise to the expenditure, rather than on chronological project phases. Under this approach, only expenditure that is directly attributable to development activities is eligible for capitalisation when the six recognition criteria for internally generated intangible assets are met. In agile software development, research activities may occur after development activities commence as part of the iterative development process. Even if development activities are followed by research activities, this approach requires the entity to identify the expenditure arising from development activities and consider whether it meets the criteria for recognition.

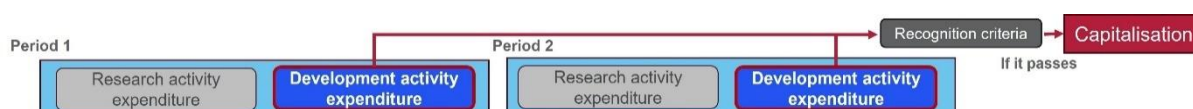


Figure 1: Option A—focused on identifying research and development activities

51. The effectiveness of this approach would depend on an entity's ability to distinguish research and development activities and to identify the costs associated with those activities. If the IASB were to pursue this approach, it could consider developing additional indicators to help entities identify research and development activities, drawing on the existing examples in paragraphs 56 and 59 of IAS 38.
52. This approach could have the following advantages and disadvantages:
- (a) its advantages include that:
 - (i) it preserves the existing outcome in IAS 38 whereby all research expenditure is expensed and only development expenditure is eligible for capitalisation. As a result, it may support continuity in practice and reduce implementation and transition costs.
 - (ii) it accommodates development approaches in which research activities may occur after development activities have commenced.
 - (iii) it focuses recognition on the nature of the underlying activities performed, rather than on project progression (a time-based

assessment) as a proxy for determining whether expenditure qualifies for recognition. Consequently, it may address stakeholder concerns about the difficulty of determining when a project enters the development phase in agile development environments.

- (b) its disadvantages include that:
- (i) identifying and attributing costs to specific activities on an ongoing basis may be operationally challenging and costly.
 - (ii) it may increase complexity because entities would need to continually assess whether the expenditure on individual activities meets the recognition criteria, rather than applying those criteria once a project has entered the development phase.
 - (iii) developing sufficiently robust descriptions of research and development activities that can be applied consistently across diverse development environments may be challenging. This could result in diversity in practice and reduce comparability.
 - (iv) this approach assumes the primary challenge in applying the existing requirements to agile development is determining when a project enters the development phase for agile development. However, if entities remain unable to reliably distinguish between research activities and development activities, the approach may be difficult to apply in practice and may not achieve its intended objective.

Option B

53. This approach retains the current two-phase model. Because agile software development is iterative, research activities can occur throughout the development process, including after the development phase—as determined for the project—has commenced. This approach would classify expenditure incurred after the commencement of the project-level development phase as development expenditure, even when the expenditure relates to research activities. The expenditure incurred

during the development phase would be eligible for capitalisation when the six recognition criteria for internally generated intangible assets have been met.

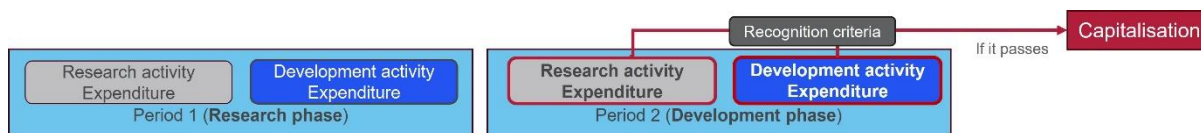


Figure 2: Option B—Retaining the two-phase model with additional guidance

54. The objective of this approach is to retain the current time-based distinction between the research and development phases while accommodating agile development. This would be achieved by providing guidance on when the development phase commences in agile software development projects and treating all subsequent expenditure after the development phase commences as development expenditure, including expenditure related to research activities.
55. If the IASB were to pursue this approach, it could consider providing additional guidance on determining when the development phase commences in an agile development environment. The IASB would also need to consider the implications of treating all subsequent expenditure after the development phase commences as development expenditure even if some of the underlying activities would otherwise be characterised as research activities.
56. This approach could have the following advantages and disadvantages:
 - (a) its advantages include that:
 - (i) it is the closest to existing IAS 38 requirements. Therefore, it may minimise disruption for entities that apply more traditional development methodologies.
 - (ii) it may reduce operational complexity compared with Option A, because determining when a project enters the development phase is a one-time judgement, rather than requiring an ongoing assessment of whether the expenditure on individual activities is related to research or development.
 - (b) its disadvantages include that:

- (i) it may not fully respond to preparers' feedback that determining when a project enters the development phase can be challenging. Although supplemented by additional guidance, the approach continues to rely on a phase-based distinction that may not fully reflect how development activities are performed in practice.
- (ii) it depends on the development of robust guidance for identifying when an agile software development project has entered the development phase, despite stakeholder feedback indicating such a determination can be difficult. Clarifications on determining the unit of account at a project-level (see paragraphs 30–38) could help entities make the determination of when the project enters the development phase, but this may not be sufficient on its own.
- (iii) it could result in capitalisation of expenditure that is substantively research in nature. Consequently, the IASB may need to reconsider the existing IAS 38 requirement that all research expenditure is expensed when incurred.

Option C

57. This approach suggests removing the current distinction between research and development phases (and research and development activities) and focuses solely on whether expenditure incurred developing internally generated intangible assets meets the six recognition criteria.

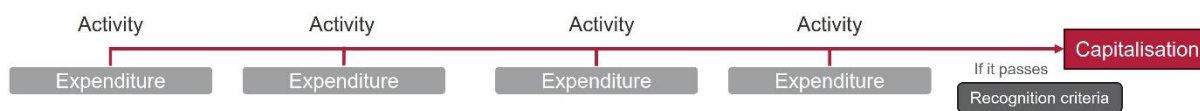


Figure 3: Option C—Removing the requirements to distinguish between the research and development phases

58. This approach could have the following advantages and disadvantages:

- (a) its advantages include that:
 - (i) it may provide a simpler recognition model by focusing on the core judgement of when the criteria for recognition as an asset are met,

rather than requiring entities to determine when a project transitions from the research phase to the development phase. This could simplify application and reduce the costs associated with assessing project progression. The approach would focus on when an intangible asset should be recognised rather than on distinguishing between research and development phases and move IAS 38 towards a standard that can more broadly apply to internally generated intangible assets and their development rather than being anchored in the distinction between research and development.

- (ii) it may better reflect agile development characteristics in which activities are often iterative and do not progress through clearly identifiable research and development phases.
 - (iii) both this approach and the current requirements in IAS 38 permit capitalisation only when the six recognition criteria for internally generated intangible assets are met. Because expenditure on research activities would generally be unlikely to meet those criteria, the accounting outcomes under this approach may not differ significantly from those under the current requirements for entities that develop intangible assets using traditional methodologies.
- (b) its disadvantages include that:
- (i) removing the requirement to identify a research phase could increase the risk of premature capitalisation because expenditure that would previously have been characterised as research expenditure could become eligible for capitalisation.
 - (ii) feedback from users suggests that information distinguishing research from development expenditure, if provided by entities, can be useful for understanding the nature of an entity's spending and assessing the extent to which such expenditure is expected to generate future economic benefits. Although the current requirements do not require entities to provide this information, removing the distinction between

research and development could reduce the availability of such information because it may become difficult or impractical to obtain the information about the nature of the underlying expenditure.

- (iii) this approach would place greater reliance on the six recognition criteria for internally generated intangible assets as the primary mechanism for determining whether expenditure is recognised as an asset. Consequently, it would increase the importance of ensuring that those criteria are sufficiently robust and operational and capable of being applied consistently.

Initial staff leaning

59. We believe that removing the distinction between the research phase and the development phase (Option C) may better reflect modern development methodologies such as agile software development, for the following reasons:
- (a) the distinction between research and development is based on broad descriptions² and limited examples about each activity³ in IAS 38. As a result, the descriptions may be difficult to apply consistently and may provide limited assistance in determining when a project has entered the development phase or in enhancing the usefulness of information provided about internally generated intangible assets.
 - (b) in applying IAS 38, recognition of internally generated intangible assets ultimately depends on whether the six recognition criteria are met. Therefore, removing the requirement to distinguish between the research and development phase may not result in fundamental changes to the application of

² Paragraph 8 of IAS 38 defines research as 'original and planned investigation undertaken with the prospect of gaining new scientific or technical knowledge and understanding' and development as 'the application of research findings or other knowledge to a plan or design for the production of new or substantially improved materials, devices, products, processes, systems or services before the start of commercial production or use'. Although the terms 'research' and 'development' are defined, the terms 'research phase' and 'development phase' have a broader meaning for the purpose of this Standard (paragraph 52 of IAS 38).

³ See paragraphs 56 and 59 of IAS 38.

the recognition requirements in many cases, including for intangible assets developed using traditional methodologies.

- (c) modern development methodologies, including agile software development, often involve iterative and continuous development activities that make the distinction between the research phase and the development phase increasingly difficult to apply in a meaningful way.
- (d) if users need information about the distinction between research and development expenditure to assess the uncertainty related to future economic benefits, the IASB could consider whether additional disclosure requirements could meet those information needs. For example, it could consider requiring disclosure based on alternative indicators of development progress, such as technical readiness levels—which are a scale for assessing how mature a technology is, meaning how far it has progressed from an initial idea to a technology demonstrated and proven in actual operation.

Six recognition criteria for internally generated intangible assets

Stakeholder challenges

60. During our outreach, a few stakeholders indicated that it is challenging to make the judgements required to apply the six recognition criteria for capitalising development costs, particularly when assessing technical feasibility and demonstrating probable future economic benefits. We also received feedback that applying the technical feasibility criterion can be more complex in agile development than in traditional development.

Our responses

61. Our outreach suggests that some stakeholders experience challenges in applying the six recognition requirements for internally generated intangible assets, in particular the technical feasibility requirement. However, we think that many of the concerns

raised may relate more to determining the appropriate unit of account than to assessing technical feasibility itself. For example, assessing whether an intangible asset is expected to be completed may be difficult without first identifying the appropriate boundaries of the item being assessed (that is, the unit of account).

62. We believe that, once an appropriate unit of account has been identified, assessing whether an intangible asset can be completed may not necessarily be more difficult in agile software or other modern development methodologies than in traditional development approaches. If the challenges identified are not unique to agile software development but instead reflect broader challenges in applying the six recognition requirements for internally generated intangible assets, it could be better to address potential improvements to those requirements in the work stream considering a broader review of the existing recognition requirements.
63. During our in-depth conversations, stakeholders also highlighted several other characteristics of agile software development that can make applying the requirements in IAS 38 challenging:
- (a) a significant proportion of development efforts may result in features that are abandoned or fail at an early stage of the development process (a so-called ‘fail fast’ approach);
 - (b) technical feasibility is often not established until a relatively late stage of the development process; and
 - (c) project objectives are often broad or aspirational at the outset and become more refined as development progresses.
64. We think these characteristics, particularly the prevalence of abandoned features and the initially aspirational nature of project objectives, reflect a high degree of uncertainty about whether an intangible asset will be successfully completed. Such uncertainty may support recognising expenditure as an expense rather than as an asset.
65. The underlying causes of the challenges associated with the six recognition criteria and whether they are specific to agile software development remains unclear.

Therefore, further consultation and analysis would be helpful to better understand the nature and extent of those challenges.

66. Although we have not yet identified the underlying causes of those challenges with sufficient certainty, we could consider the following potential approaches to improving the recognition criteria:
- (a) adding application indicators—this approach would supplement the existing recognition criteria with non-exhaustive indicators to support more consistent application. For example, drawing on concepts used in the updated US GAAP requirements for internal-use software, we could suggest indicators such as novel or unproven functionality, significant technical risks and stable performance requirements to help entities assess when the recognition criteria have been satisfied without changing the underlying recognition model.
 - (b) targeted revisions to the recognition criteria—this approach would involve revising specific criteria, especially technical feasibility and probable future economic benefits, to better reflect how projects progress and how uncertainty is resolved in practice. For example, the criteria could be reframed using concepts such as significant technical and commercial uncertainty, intended functionality and project maturity, while retaining the existing recognition objective and underlying principles.
 - (c) reorganising the six recognition criteria—this approach would involve regrouping the existing criteria into core recognition criteria, such as probable future economic benefits and reliable measurement and criteria used to assess whether those core criteria have been met. Such a reorganisation could help entities better understand the conceptual purpose of the criteria and how they operate within the overall recognition framework.

Next steps for exploring the agile software development test case

67. We think it is important to first agree on the directional questions discussed in this paper, select the options to pursue further—including on the unit of account, the two-

phase model and the recognition criteria for internally generated intangible assets—and understand their implications.

68. As part of this work, we plan to discuss the options and their potential implications, including for a wider range of intangible assets, with consultative groups and other stakeholders. We will also seek to identify areas that may require further exploration, including whether there are agile software development specific matters concerning the six recognition criteria for internally generated intangible assets, before returning to the IASB in early 2027.
69. When we have more clarity on the preferred options for the questions above, we would then consider:
- (a) other challenges raised by stakeholders in relation to agile software development, including whether improvements could be made:
 - (i) to the requirements for subsequent expenditure to help entities distinguish maintenance cost from enhancement costs;
 - (ii) to certain aspects of the measurement requirements in IAS 38, including when to begin amortisation; and
 - (iii) to help entities assess whether previously recognised development costs should be impaired when subsequent expenditure is recognised as an asset.
 - (b) how any improvements resulting from the work described in paragraphs 67–68 and 69(a) could be incorporated into IAS 38 if no unintended consequences are identified.

Questions for IASB members

Questions for IASB members

1. Do you have any questions or comments on the staff's analysis of the unit of account, as set out in paragraphs 15–38 of this paper?
2. Do you have any questions or comments on the staff's analysis on the two-phase model that distinguishes between research and development, as set out in paragraphs 39–59 of this paper?
3. Do you have any questions or comments on the staff's analysis on the six recognition requirements for internally generated intangible assets, as set out in paragraphs 60–66 of this paper?
4. Do you have any questions or comments on the staff's suggested next steps, as set out in paragraphs 67–69?

Appendix A—Summary of FASB Accounting Standards Update 2025-06

- A1. In September 2025, the Financial Accounting Standards Board (FASB) issued [Accounting Standards Update 2025-06](#), *Intangibles—Goodwill and Other—Internal-Use Software (Subtopic 350-40): Targeted Improvements to the Accounting for Internal Use Software* (ASU 2025-06). The main amendments in ASU 2025-06 include:
- (a) removing all references to prescriptive and sequential software development project stages. As a result, an entity will be required to begin capitalising software costs when both of the following occur:
 - (i) management has authorised and committed to funding the software project; and
 - (ii) it is probable that the project will be completed and the software will be used to perform the function intended (referred to as the ‘probable-to-complete recognition threshold’).
 - (b) clarifying that if there is significant uncertainty associated with the development activities of the software (referred to as ‘significant development uncertainty’), the probable-to-complete recognition threshold is not considered to be met until the uncertainty has been resolved.
 - (c) introducing two factors that indicate that significant development uncertainty exists:
 - (i) novel or unproven software—the software being developed has technological innovations or novel, unique, or unproven functions or features and the uncertainty related to those technological innovations, functions, or features, if identified, has not been resolved through coding and testing; and

-
- (ii) significant performance requirements—the significant performance requirements of the software have not been identified, or the identified significant performance requirements continue to be substantially revised.
- A2. The unit of account for applying the capitalisation requirements in Subtopic 350-40 to internal-use software is a ‘software project’. However, a software project is not defined in Subtopic 350-40 and the amendments in ASU 2025-06 do not specifically define what constitutes a software project. The FASB decided not to address the unit of account in applying Subtopic 350-40 because providing guidance could change practice or limit the judgement that is currently allowed. In the FASB’s view, entities should continue to apply judgment in determining what constitutes a software project on the basis of its specific facts and circumstances (paragraphs BC61–BC63 of ASU 2025-06).
- A3. The amendments in ASU 2025-06 are effective for annual reporting periods beginning after 15 December 2027 and interim reporting periods within those annual reporting periods. Early adoption is permitted.

Appendix B—Fact pattern explored in this paper***Fact pattern—video game development using an agile methodology***

- B1. A video game developer uses an agile methodology to develop games. On 15 January 20X1, management approves funding to develop a new online adventure game. The development roadmap includes five planned features:
- (a) feature 1: Core Gameplay;
 - (b) feature 2: Quest Tracking;
 - (c) feature 3: Multiplayer;
 - (d) feature 4: Marketplace; and
 - (e) feature 5: Story Engine.
- B2. Features 1 and 2 are based on technology used in previous games and largely replicate functionality available in earlier releases. Feature 1 provides the game's core mechanics, while feature 2 enables players to track quests, objectives and their progress through the game. By 15 January 20X1, the relevant research activities for features 1 and 2 have already been completed. Although those features could be completed and deployed before the other planned features, management does not intend to release or commercialise the game based solely on them because they do not provide meaningful differentiation in functionality from existing products.
- B3. Feature 3 allows players to connect and play together online. Although the entity has not previously implemented the proposed networking architecture, management believes it can be developed using existing knowledge and resources. The approved business case identifies features 1–3 together as the minimum functionality required for an initial release. Those features are intended to operate together as an integrated game experience and provide the combined functionality necessary for the initial release. Management expects this combination of functionality to provide the first meaningful differentiation from previous releases and the first point at which it believes future economic benefits can be generated from the game.

-
- B4. Feature 4 allows players to trade in-game items. The entity has limited experience with this functionality and key requirements continue to be evaluated. Feature 5 dynamically adapts the game's narrative based on player behaviour and represents a new type of functionality for the entity. Significant uncertainty exists regarding whether features 4 and 5 can be successfully developed. Both features are included in the approved project roadmap. However, management intends to release the game once features 1–3 are completed and operating as intended while continuing to develop features 4 and 5.
- B5. By 30 June 20X1, the entity completes the design, prototyping and testing activities for feature 3 and concludes that the functionality can be completed and integrated with features 1 and 2 as planned. Features 4 and 5 remain subject to significant technical and commercial uncertainty.
- B6. On 15 October 20X1, the entity releases a version of the game incorporating features 1–3, and starts earning revenue from this initial release. Development of features 4 and 5 continues.
- B7. On 15 December 20X1, the entity concludes that features 4 and 5 cannot be completed within acceptable technical and commercial parameters and abandons their development.

View A—Single unit of account based on the overall project

- B8. Under View A, the unit of account is linked to features 1–5 because they were approved as part of one project and intended to contribute to a single product.

Analysis

- B9. Although features 1 and 2 are based on established technology and feature 3 is demonstrated to be achievable on 30 June 20X1, features 1–5 are assessed together as a single unit of account. Consequently, the uncertainty associated with features 4 and 5 prevents the recognition criteria from being met for the project as a whole. By the time features 4 and 5 are abandoned on 15 December 20X1, a version of the game incorporating features 1–3 has already been released and is generating revenue.

However, because all five features are assessed together, all development expenditure relating to the game has been recognised as an expense and no intangible asset has been recognised.

View B—Separate units of account based on distinct functionality

- B10. Under View B, the unit of account is linked to each of features 1–5 because each feature provides distinct functionality. Accordingly, each feature is assessed separately.

Analysis

- B11. Because each feature is assessed separately, the uncertainty associated with features 4 and 5 does not affect the recognition assessment for features 1–3. Eligible development expenditure relating to features 1 and 2 begins to be recognised on 15 January 20X1 because the relevant research activities for those features have been completed, and those features provide distinct functionality and are based on established technology that the entity has successfully deployed in previous releases. Eligible development expenditure relating to feature 3 begins to be recognised on 30 June 20X1 following completion of the design, prototyping and testing activities. Features 4 and 5 never satisfy the recognition criteria and are abandoned on 15 December 20X1.
- B12. Under this view, eligible development expenditure relating to features 1 and 2 begin to be recognised on 15 January 20X1 even though management does not intend to release or generate future economic benefits from those features separately. When the game incorporating features 1–3 is released on 15 October 20X1, amortisation begins for the recognised assets relating to features 1–3.

View C—Unit of account based on the smallest aggregation of functionality expected to generate future economic benefits

- B13. Under View C, the unit of account is linked to features 1–3 because together they represent the smallest aggregation of functionality from which management expects to generate future economic benefits.

Analysis

- B14. Although features 1 and 2 are based on established technology and provide distinct functionality, management does not intend to release or expect to generate future economic benefits from those features separately. The addition of feature 3 provides the first meaningful differentiation from previous releases and represents the first point at which management expects to generate future economic benefits.
- B15. On 15 January 20X1, the recognition criteria are not met because the smallest aggregation of functionality from which management expects to generate future economic benefits comprises features 1–3, and feature 3 remains subject to technical uncertainty. On 30 June 20X1, following completion of the design, prototyping and testing activities for feature 3, eligible development expenditure relating to features 1–3 begins to be capitalised. When the game incorporating features 1–3 is released on 15 October 20X1, amortisation begins for the recognised asset.
- B16. Features 4 and 5 would provide additional functionality. However, features 1–3 represent the appropriate aggregation of functionality because they are the minimum functionality required for management to generate future economic benefits from the game. Features 4 and 5 are incremental to, rather than necessary for, that aggregation of functionality and are therefore assessed separately. Because the recognition criteria are never met for features 4 and 5, related expenditure is recognised as an expense.