



Working Draft: Epic Definition & Evaluation Guide

If you are using the Epic Brief and this Guide, your feedback is welcome. Always looking to improve our tools and techniques! Please reach out and let us know what you think or if you'd like to discuss in more detail.

Jeff Howey
The Agile Alchemist
jhowey@thinklouder.com

Epic Brief Definition & Evaluation Guide

Intake

1. **Definition and Alignment:**

- 1.1. Ensure the Epic's narrative is articulated effectively, keeping in mind the potential persona-based or platform-based context. The narrative should be value-driven and able to communicate the customer need.
- 1.2. Confirm that the Epic aligns with the overall product or business strategy.

2. **Quantitative Criteria:**

- 2.1. Assign the Epic a T-shirt size as an initial estimate of its scale or effort.
- 2.2. Position the Epic in the backlog, based on perceived value.

3. **Qualitative Criteria:**

- 3.1. Validate that the Problem Statement in the Epic is clearly defined and specific.
- 3.2. Make certain that the Epic is focusing on solving a customer problem or fulfilling a customer need.

4. **Advice:**

- 4.1. Foster cross-functional collaboration: Facilitate input and dialogue with stakeholders, including the product manager, solution architect, and delivery teams. This ensures a comprehensive understanding of the Epic.
- 4.2. Be open to refinement: During this stage, the purpose is to capture the essence of the Epic, which will be progressively refined and elaborated in subsequent steps. However, strive for as much clarity and specificity as possible, to lay a solid foundation for later stages.
- 4.3. Stay flexible: The nature of the Epic might evolve as it moves through each stage, influenced by new insights, stakeholder feedback, or market dynamics. Remain open to re-defining or even discarding an Epic if it no longer aligns with strategic goals or customer needs.

5. **Artifacts to Update:**

- 5.1. Update the product backlog with the new Epic.
- 5.2. Incorporate the Epic in the product roadmap, marking its alignment with strategic goals.
- 5.3. Update Agile Lifecycle Management tools with the Epic's details, if applicable.

6. In the Epic Brief, ensure that the Problem Statement, Description/Narrative, Acceptance Criteria, Success Measures, and Preliminary Risks, Assumptions, Constraints, and Dependencies are drafted with as much clarity as possible at this early stage. These components will be subject to iterative refinement in later stages of the Epic lifecycle.

Solution Design

1. **Definition and Alignment:**
 - 1.1. Ensure that the Epic continues to align with the overall product or business strategy.
 - 1.2. Confirm that the narrative remains consistent with the Persona-based or Platform-based context, maintaining its focus on delivering customer value.
2. **Quantitative Criteria:**
 - 2.1. Validate the Epic's T-shirt size with the proposed solution design in mind, adjusting the estimate as necessary.
 - 2.2. Articulate a Solution Hypothesis which can guide the solution design process.
 - 2.3. Identify and list any dependencies the Epic may have on other Epics or systems.
 - 2.4. Extract the backbone stories from the Epic and align them with the proposed solution design.
3. **Qualitative Criteria:**
 - 3.1. Verify that the Solution Design in the Epic is clearly defined and adequately addresses the problem statement.
 - 3.2. Check that the Epic has a tangible and feasible path to solving the identified customer problem or fulfilling the customer need.
4. **Advice:**
 - 4.1. Collaborate with cross-functional teams: Engage solution architects, UX/UI designers, engineers, and other stakeholders to ensure that the solution design is viable and aligns with the Epic's objectives.
 - 4.2. Refine and iterate: At this stage, your solution design should be viewed as a hypothesis that will be iteratively refined and validated in subsequent steps.
 - 4.3. Manage dependencies proactively: The identification of dependencies is crucial at this stage, as they may affect the Epic's feasibility, prioritization, and estimated effort.
5. **Artifacts to Update:**
 - 5.1. Update the Epic in the product backlog, incorporating any refinements in estimated size, and the addition of the solution hypothesis, and identified dependencies.
 - 5.2. Update the Epic in the product roadmap, potentially indicating it as a goal depending on its strategic significance.
 - 5.3. Update Agile Lifecycle Management tools with new details on solution design, dependencies, and backbone stories, if applicable.
6. In the Epic Brief, ensure that the Problem Statement, Description/Narrative, Acceptance Criteria, Success Measures, Solution Design, Backbone Story Map, Prioritization, Risks, Assumptions, Constraints, and Dependencies are refined as necessary at this stage, focusing specifically on the elaboration of the Solution Design. This more detailed articulation of the proposed solution will serve as a foundation for the Planning and Development steps to follow.

Planning

1. **Definition and Alignment:**
 - 1.1. Ensure that the Epic continues to align with the overall product or business strategy.
 - 1.2. Verify that the narrative remains consistent with the Persona-based or Platform-based context, focusing on delivering customer value.
2. **Quantitative Criteria:**
 - 2.1. Verify that the T-shirt size of the Epic aligns with the now more detailed solution design and the estimated effort of the Backbone Stories.
 - 2.2. Confirm the alignment of dependencies with the proposed solution design and map out their impact on the delivery of the Epic.
 - 2.3. Make sure that the Backbone Stories are well defined and have estimated efforts.
3. **Qualitative Criteria:**
 - 3.1. Ensure that the planning process is comprehensive, encompassing all crucial elements of the Epic, including dependencies, effort estimates, and the sequence of the Backbone Stories.
 - 3.2. Check that the planning reflects a feasible path to the solution, considering dependencies and estimated efforts.
4. **Advice:**
 - 4.1. Use cross-functional collaboration: Involvement of solution architects, engineers, QA, and other stakeholders is vital at this stage to ensure accurate estimates, risk assessments, and alignment on the delivery plan.
 - 4.2. Keep refining and iterating: Your plan should be adaptable, ready for changes as new information or feedback is received in subsequent steps.
 - 4.3. Proactively manage dependencies: Make sure dependencies are not only identified but also mapped in terms of their impact on the delivery schedule.
5. **Artifacts to Update:**
 - 5.1. Update the Epic in the product backlog to reflect refined effort estimates, the state of dependencies, and the detailed planning.
 - 5.2. Update the Epic in the product roadmap to show its readiness for inclusion in a future planning increment.
 - 5.3. Update Agile Lifecycle Management tools to reflect the new level of detail achieved in the planning process.
6. In the Epic Brief, ensure that the Problem Statement, Description/Narrative, Acceptance Criteria, Success Measures, Solution Design, Backbone Story Map, Prioritization, Risks, Assumptions, Constraints, and Dependencies are all refined as necessary at this stage, focusing specifically on the elaboration of the planning. This detailed plan will serve as the foundation for the upcoming Development step.

Epic Ready

1. **Definition and Alignment:**
 - 1.1. Verify that the Epic remains aligned with the overall product or business strategy.
 - 1.2. Confirm the narrative of the Epic continues to reflect a Persona-based or Platform-based focus on customer value.
2. **Quantitative Criteria:**
 - 2.1. Confirm that the Epic is broken down into refined, implementable Stories, with associated story points, ready for the next 2-3 Sprints at least.
 - 2.2. Ensure there's available capacity in the team to work on the Epic based on its priority and the estimated efforts.
 - 2.3. Ensure the Acceptance Criteria are clearly defined, specific, measurable, and actionable, offering clarity for the delivery team.
3. **Qualitative Criteria:**
 - 3.1. Ensure that the solution and the approach to its delivery are feasible and achievable within the estimated efforts and timelines.
 - 3.2. Ensure all parties involved have a shared understanding and alignment about what is being built and the value it brings.
4. **Advice:**
 - 4.1. **Prioritize Cross-functional Collaboration:** Make sure all stakeholders understand and agree on the Epic's details, including solution architects, engineers, QA, and other stakeholders. Open communication channels can help ensure that everyone is on the same page.
 - 4.2. **Validate Against Reality:** Verify the team's capacity and availability to work on the Epic, ensuring that it aligns with its priority and estimated effort.
 - 4.3. **Be Adaptable:** Even at this stage, remain open to refinements based on new information, changes in the business context, or feedback from the delivery teams.
5. **Artifacts to Update:**
 - 5.1. Update the Epic in the product backlog with its readiness status and its associated, broken down user stories.
 - 5.2. Update Agile Lifecycle Management tools with the Epic's readiness status and the detailed user stories, reflecting their readiness for development.
 - 5.3. Update the Product Roadmap and Release Plans with the Epic's readiness for the upcoming PI or 3-6 Sprints, depending on the planning horizon of the organization.
6. In the Epic Brief, the Problem Statement, Description/Narrative, Acceptance Criteria, Success Measures, Solution Design, Backbone Story Map, Prioritization, Risks, Assumptions, Constraints, and Dependencies should be well refined and provide detailed clarity for the delivery team. At this stage, all information should be clear and precise, enabling the delivery team to begin work with a clear understanding of what needs to be built and why.

Development (In Progress)

1. **Definition and Alignment:**
 - 1.1. Ensure the development work aligns with the Epic's Description/Narrative and Acceptance Criteria as refined during prior steps.
 - 1.2. Confirm the narrative of the Epic continues to be reflected in the work being performed and aligns with the Persona-based or Platform-based focus on customer value.
2. **Quantitative Criteria:**
 - 2.1. Track progress of story completion against the planned effort and capacity.
 - 2.2. Continually review and manage the Epic's Dependencies, Risks, Assumptions, and Constraints.
 - 2.3. Implement an iterative review process to validate against Acceptance Criteria and ensure the development work is aligning with the intended solution design.
3. **Qualitative Criteria:**
 - 3.1. Ensure the Epic's development continues to meet the Solution Hypothesis and Success Measures outlined in the Epic Brief.
 - 3.2. Continually validate that the development work is focusing on solving the identified Problem Statement.
4. **Advice:**
 - 4.1. Foster Collaboration: Keep an open line of communication between the delivery team and product team to address any roadblocks, resolve clarifications, and adapt to changes as required.
 - 4.2. Facilitate Regular Demos: Regularly demonstrate the increments of work completed to stakeholders for early feedback and alignment.
 - 4.3. Manage Risks and Dependencies: Proactively identify and manage risks and dependencies that may impact the Epic's development, and plan mitigation strategies if necessary.
 - 4.4. Trim the Tail: Continually review the stories to identify if sufficient value can be delivered earlier, allowing teams to shift to other priorities if applicable.
5. **Artifacts to Update:**
 - 5.1. Update the Epic's status in the product backlog and ALM tools to reflect its development progress.
 - 5.2. Document and manage Risks, Assumptions, Constraints, and Dependencies in relevant ALM tools, project management tools, or risk registers.
 - 5.3. Update the Product Roadmap and Release Plans based on the actual progress and any changes made during the development.
6. Throughout the Development process, the Product Team, including the Product Manager, Solution Architect, and others, should actively monitor and control the progress to ensure it aligns with the Epic's intended value and solution design. Regular collaboration with the Delivery Team is vital to mitigate risks, manage dependencies, and ensure alignment with the Acceptance Criteria and Solution Hypothesis.

Deployment

1. **Definition and Alignment:**

- 1.1. Confirm that the completed development work aligns with the Epic's Description/Narrative and Acceptance Criteria and solves the identified Problem Statement.
- 1.2. Ensure the product increment that's ready for deployment delivers the customer value outlined in the Epic Brief and aligns with the Persona-based or Platform-based focus.

2. **Quantitative Criteria:**

- 2.1. Ensure all Acceptance Criteria are met before deployment.
- 2.2. Verify that the Solution Hypothesis is validated with the finished product increment.
- 2.3. Confirm that the Epic's Dependencies, Risks, Assumptions, and Constraints have been effectively managed and closed out where appropriate.
- 2.4. Validate that the Success Measures are clearly defined and can be tracked post-deployment.

3. **Qualitative Criteria:**

- 3.1. Confirm that the implemented solution effectively solves the identified Problem Statement.
- 3.2. Ensure the final product increment meets the quality standards and is ready for deployment.

4. **Advice:**

- 4.1. Engage in Stakeholder Demos: Before deployment, hold a demo for stakeholders to showcase the final product increment and get approval for deployment.
- 4.2. Plan for Deployment: Consider deployment strategies (like blue/green or canary deployments) to reduce risks associated with deployment. Plan for any necessary rollback measures.
- 4.3. Prepare Support Documentation and Training: Ensure any necessary user documentation, support guides, or training materials have been prepared and are ready for dissemination post-deployment.
- 4.4. Post-Deployment Reviews: Schedule reviews to gather feedback and learnings after deployment, and plan for any necessary adjustments or iterations on the Epic.

5. **Artifacts to Update:**

- 5.1. Update the Epic's status in the product backlog and ALM tools to reflect its deployment.
- 5.2. Document learnings, challenges, and successes in relevant project management or ALM tools for future reference.
- 5.3. Update the Product Roadmap and Release Plans based on the deployed Epic.
- 5.4. Prepare and disseminate any necessary user guides, support documentation, or training materials.

6. The Deployment step is a critical stage where the product increment is finally delivered to users. The focus here should be on ensuring a smooth deployment process and the successful delivery of customer value as intended in the Epic Brief. It's also a point to celebrate the team's achievement while preparing to measure and review the Epic's impact post-deployment.

Validation

1. **Definition and Alignment:**
 - 1.1. Ensure that the deployed product increment fulfills the Epic's Description/Narrative and Acceptance Criteria, effectively addresses the Problem Statement, and delivers the intended customer value outlined in the Epic Brief.
 - 1.2. Validate that the implemented solution aligns with the Persona-based or Platform-based focus and contributes to the desired Business Outcomes.
2. **Quantitative Criteria:**
 - 2.1. Measure and confirm that the Success Measures defined in the Epic Brief are being met post-deployment.
 - 2.2. Track customer engagement and feedback metrics to validate that the product increment delivers the desired customer experience and value.
 - 2.3. Assess the financial performance and ROI of the Epic.
3. **Qualitative Criteria:**
 - 3.1. Collect customer feedback to validate the quality and usability of the product increment, and its effectiveness in addressing the Problem Statement.
 - 3.2. Gather team feedback and conduct a retrospective to identify what went well and potential improvements for future Epics.
4. **Advice:**
 - 4.1. Monitor User Feedback: Regularly collect and review user feedback to validate the success of the product increment and identify areas for potential improvements.
 - 4.2. Conduct Retrospective: Hold a team retrospective to reflect on the overall process, learn from the experience, and identify ways to improve future Epics and deployments.
 - 4.3. Post-Validation Review: Conduct a review with stakeholders to discuss the outcome, the success measures, and the next steps. This should include a reassessment of the Epic's alignment with strategy and potential follow-up actions.
5. **Artifacts to Update:**
 - 5.1. Document the validation findings and lessons learned in relevant project management or ALM tools for future reference.
 - 5.2. Update the Product Roadmap and Strategic Plan based on the learnings and impact of the deployed Epic.
 - 5.3. Prepare a post-deployment report summarizing the Epic's performance against the defined Success Measures, and share it with the relevant stakeholders.
6. Validation is the final step where the impact of the deployed Epic is evaluated against its intended outcomes. This step offers invaluable insights to drive continuous improvement in the product and the process. It not only provides a chance to celebrate the successes but also to learn from the experience and refine future Epics and strategies.