Core Project Evaluation Criteria

Technical Steering Committee

INTERSECT™

Core Scope

We define our scope as the continuity of Cardano;

Continuity is the unbroken and consistent existence or operation of something over time.

Cardano “Continuity” is defined as any approved technical services needed to build or fix current and future areas of the Cardano blockchain. Continuity in this definition can include new development, such as the CIP-1694 governance software, upgrades to components or improvements in testing.

To further help this definition we have outlined specific areas which would fall under consideration for TSC endorsement, funding and deliver: Cardano Maintenance, Cardano Development, Market Drivers and Appeal.

At any given time the scope under delivery of the TSC will incorporate these elements. The total makeup of scope will vary over time, with those nearer the core having weighted priority.

Scope Visualization

● **Cardano Maintenance**

Required to keep current mainnet node technology and dependencies safely running and secure

● **Cardano Development**

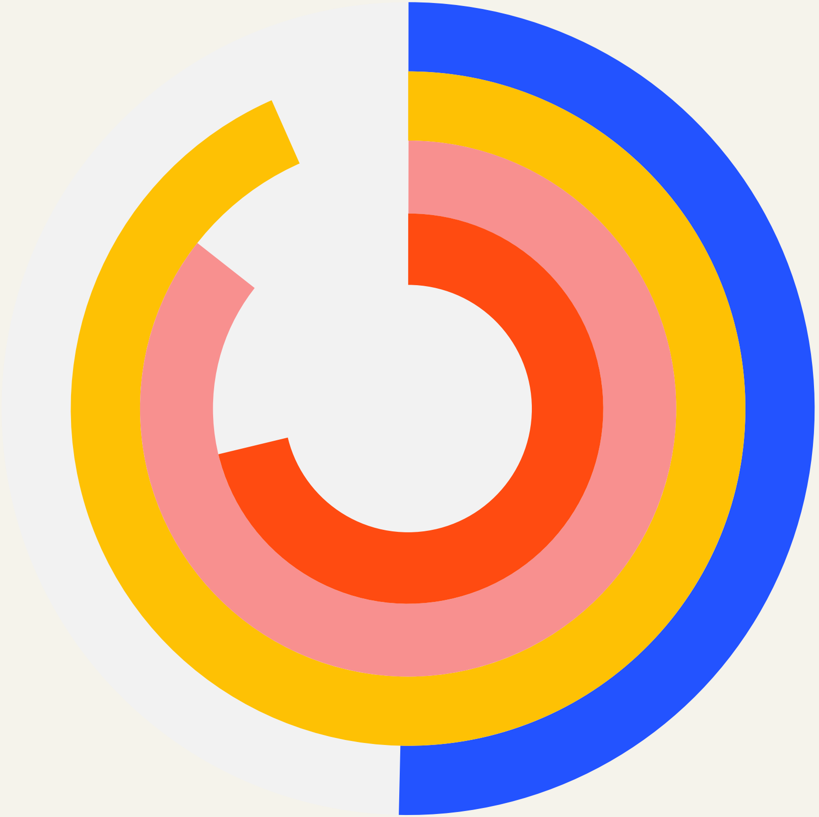
Enhancement to core features such as Ledger, Consensus, Networking and the Node itself

● **Market Drivers**

Technology deemed required to keep Cardano competitive amongst other blockchains

● **Community Appeal**

Technology with core dependencies and which has mass appeal from the ecosystem, as shown through prioritization feedback and would benefit from delivery coordination



Note there are other ecosystem funding opportunities which may measure and prioritize these factors differently.

Scope continued

All core infrastructure must

- Be **open source**
- Be **transparent** and **open to feedback**, **adjustment and re-prioritization**
- **Not create long term financial benefit** to a specific person(s) or organization(s)
- Provide **low-barrier to inclusion** or educational opportunities for the community

1. Importance

- **Is the project maintaining essential core infrastructure**

- **Is the project as a whole needed**
  - does it align with the community-agreed roadmap
  - are all sub-projects needed - if not, can they be split off/removed

- **How well supported is the project**
  - general evidence of interest
  - interest from specific communities

- **Does the project overlap with other proposals**
  - in what ways
  - should both proposals be considered

2. Technical Quality

- **Does the project as a whole make technical sense**
  - Is the work technically feasible
  - Are there elements that may be infeasible - if so, there should be “gating”

- **Are all outcomes defined precisely**
  - How can the success of the project be judged
  - Have SRLs (or equivalent) been used to precisely identify the initial and final states of development

- **Are all milestones and deliverables explicit and sensible**
  - Are all SRL stages necessary
  - Are milestones/deliverables provided that map to each SRL in the TSC chart, where needed
  - Do sub-projects link together properly

INTERSECT™

2. Technical Quality (Continued)

- **To complete the project will there need to be adjustments to the current (and proposed) structure of Cardano L1**
  - Is this included in the project proposal

- **Have all risks been identified and mitigations considered**
  - both technical and social
  - including security and performance

- **Will the project deliver a system that is robust, reliable and resilient**
  - How is this determined

- **Is security properly considered**
  - Will necessary security audits be carried out
  - Is security considered as a primary rather than secondary goal

2. Technical Quality (Continued)

- **Are all dependencies identified**
  - Including with other teams/projects

- **Have integration, rollout and deployment been properly considered**
  - How will they be addressed if not

- **Is the proposal consistent with the current constitution**
  - If not, what constitution changes will be needed for deployment

INTERSECT™

3. Appropriateness of Effort

- **Has too much effort been allocated**
    - Is the work too simple for the requested resources

- **Is too little effort allocated**
    - Is the project likely too ambitious and therefore likely to fail

- **For complex projects, each sub-project needs to be considered separately**

- **Has suitable thought been given to how this will be integrated into the node**

- **Will the development have knock on effects in tooling, user experience etc**

- **Is it the intention that the community will take on maintenance in the long term**
    - If so, this needs to be considered in future budgets

4. Team Quality

- **Does the team collectively have the skills and experience that are needed to carry out the work**
    - Individual developers matter, not just the organisation

- **If there is expertise that is needed that the team does not have has it been clearly identified**
    - How do they plan to source the needed expertise
    - Do they need help against specialist resources that exist in the community or wider

- **Can the team attract additional members/replace members if necessary**
    - It may be necessary to cover gaps
    - The project could be fragile if critical team members cannot be replaced (“bus factor”)

- **What projects has the team carried out before**
    - Have they been completed to a high standard
    - Are they similar in scope
    - Are they of a similar scale or is the team stretching itself

5. Value for Money

- **Does the project represent good value for money**
    - Prioritising technical quality importance and team quality over pure financial cost
    - A fair return is reasonable, an excessive return is not

- **What, if any, adjustments are needed to make the project provide good value**
    - Is it reasonable to expect these from the supplier

6. Financial

- Are all costs properly identified and explained

- Will the project be properly audited to international standards

- Will it be possible to verify that effort has been properly deployed

- Will unspent funds be returned to the community, or will they be absorbed by the supplier

- Does the supplier have sufficient depth to fund the cash flow requirements

7. Commitment

- **Is the supplier fully committed to the project**
  - What happens if they pull out
  - Have they shown a willingness for long-term commitment

- **If this is a critical project, is some other supplier able to replace them if they do fail/withdraw**
  - This may incur additional expense for the community

8. Essential Changes

- **Do essential changes need to be made to the project**
  - What are they

- **Are the essential changes possible**
  - If not the project should not be accepted

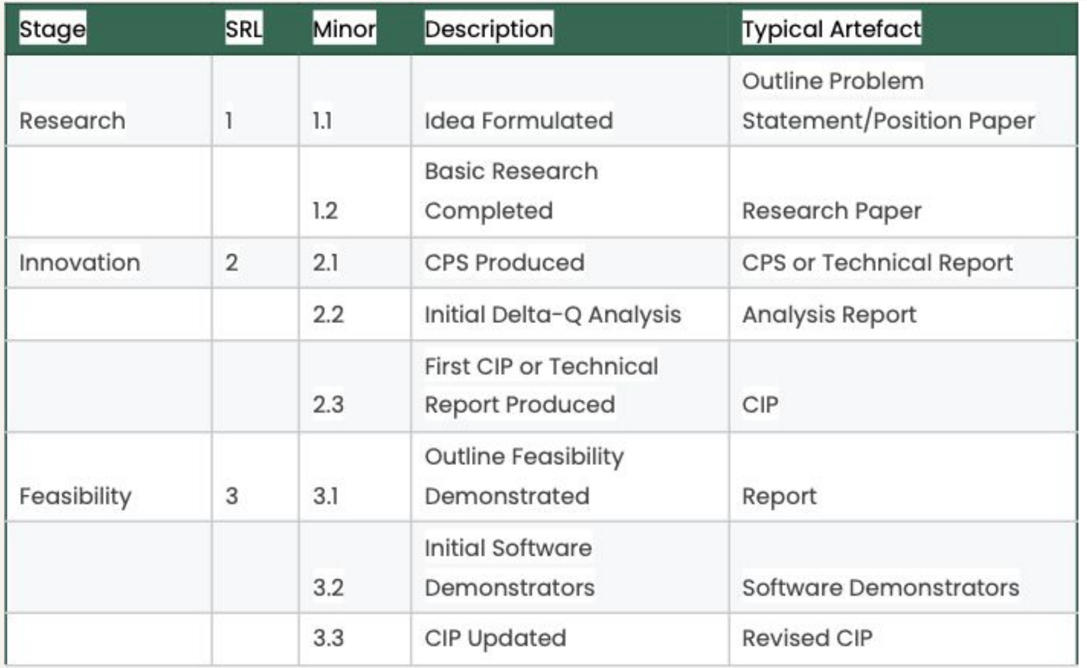
INTERSECT™

8. Essential Changes

- **Do essential changes need to be made to the project**
  - What are they

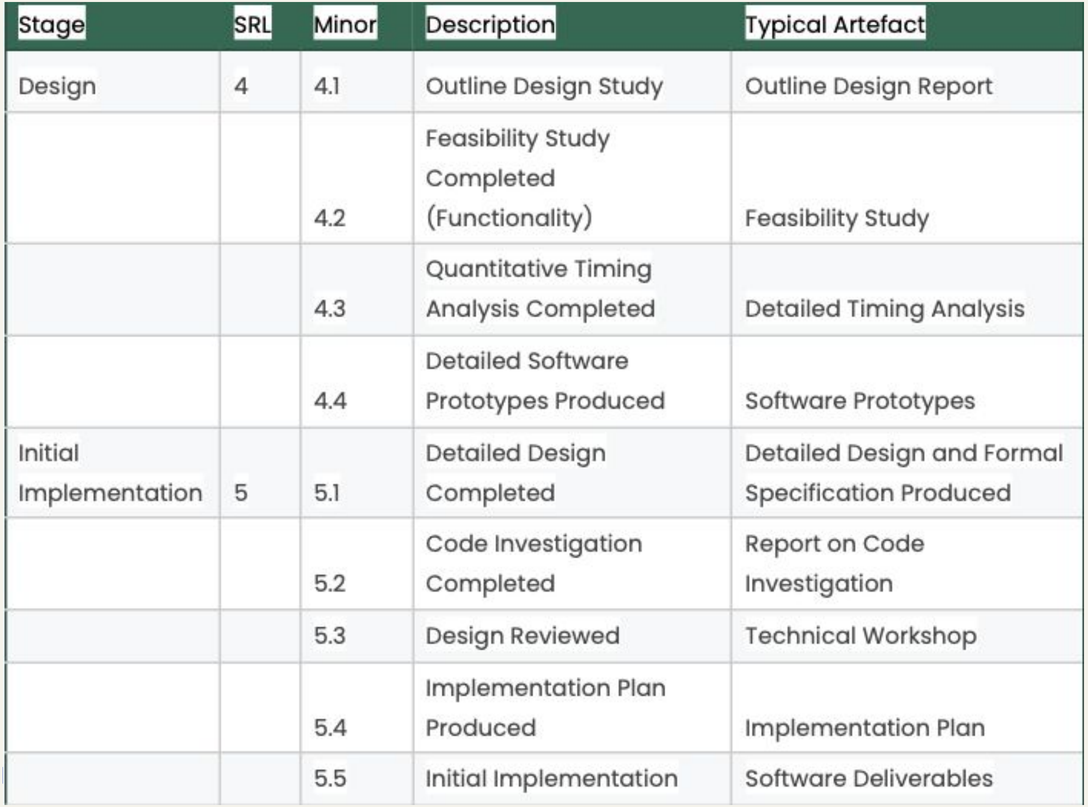
- **Are the essential changes possible**
  - If not the project should not be accepted

Software Readiness Levels (SRLs): 1-3



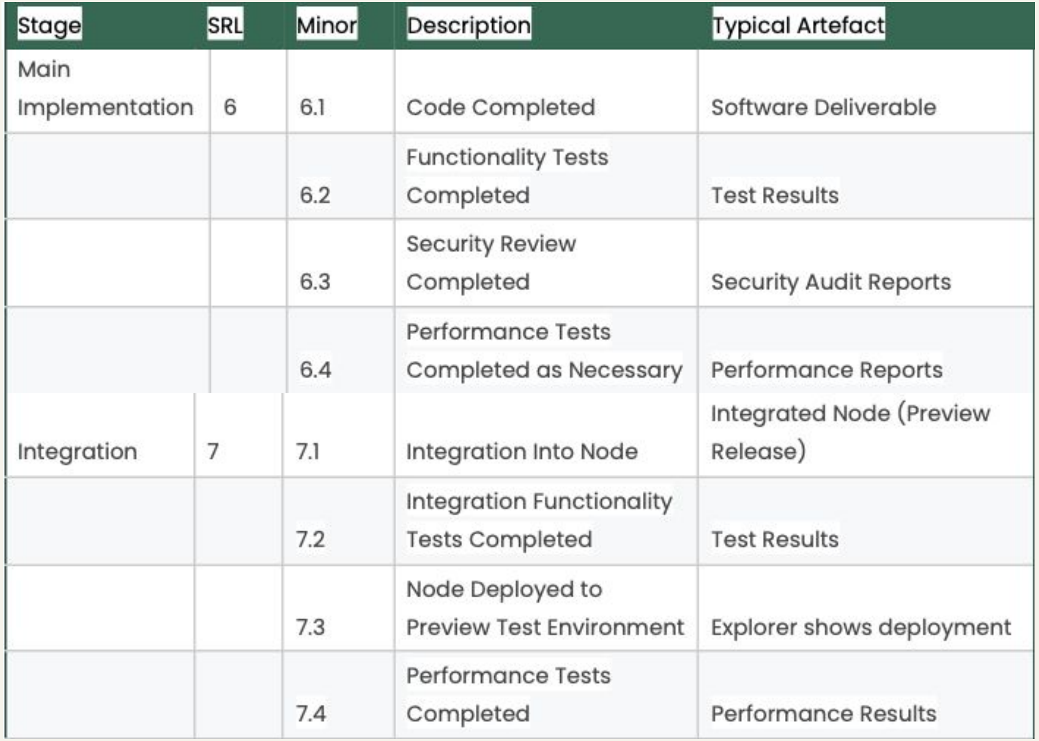
Stage	SRL	Minor	Description	Typical Artefact
Research	1	1.1	Idea Formulated	Outline Problem Statement/Position Paper
		1.2	Basic Research Completed	Research Paper
Innovation	2	2.1	CPS Produced	CPS or Technical Report
		2.2	Initial Delta-Q Analysis	Analysis Report
		2.3	First CIP or Technical Report Produced	CIP
Feasibility	3	3.1	Outline Feasibility Demonstrated	Report
		3.2	Initial Software Demonstrators	Software Demonstrators
		3.3	CIP Updated	Revised CIP

Software Readiness Levels (SRLs): 4-5

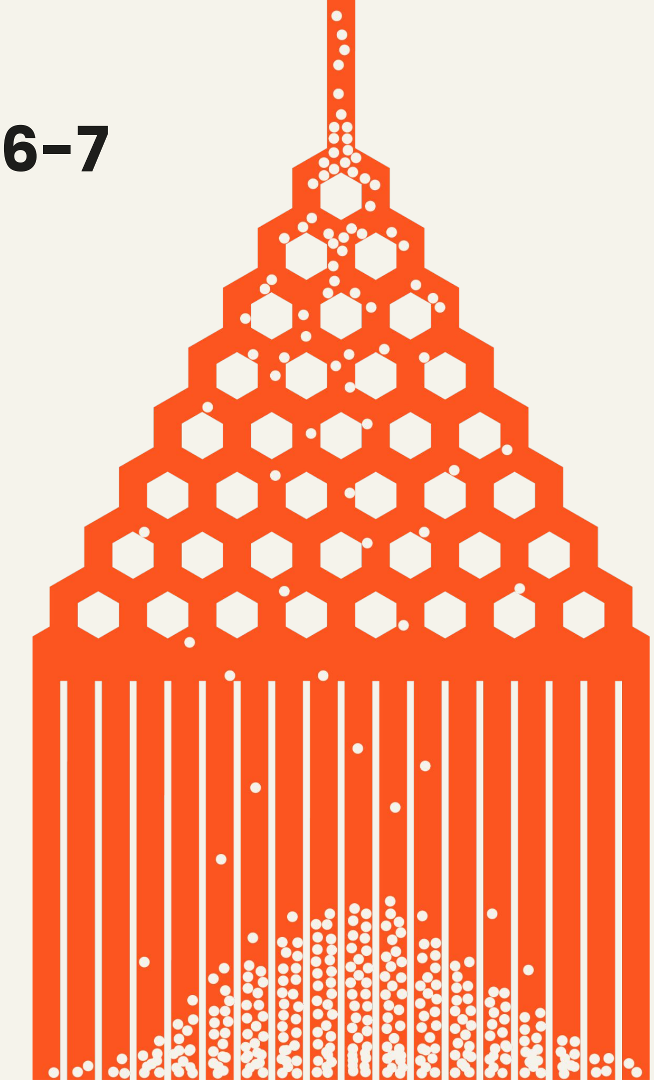


Stage	SRL	Minor	Description	Typical Artefact
Design	4	4.1	Outline Design Study	Outline Design Report
		4.2	Feasibility Study Completed (Functionality)	Feasibility Study
		4.3	Quantitative Timing Analysis Completed	Detailed Timing Analysis
		4.4	Detailed Software Prototypes Produced	Software Prototypes
Initial Implementation	5	5.1	Detailed Design Completed	Detailed Design and Formal Specification Produced
		5.2	Code Investigation Completed	Report on Code Investigation
		5.3	Design Reviewed	Technical Workshop
		5.4	Implementation Plan Produced	Implementation Plan
		5.5	Initial Implementation	Software Deliverables

Software Readiness Levels (SRLs): 6-7

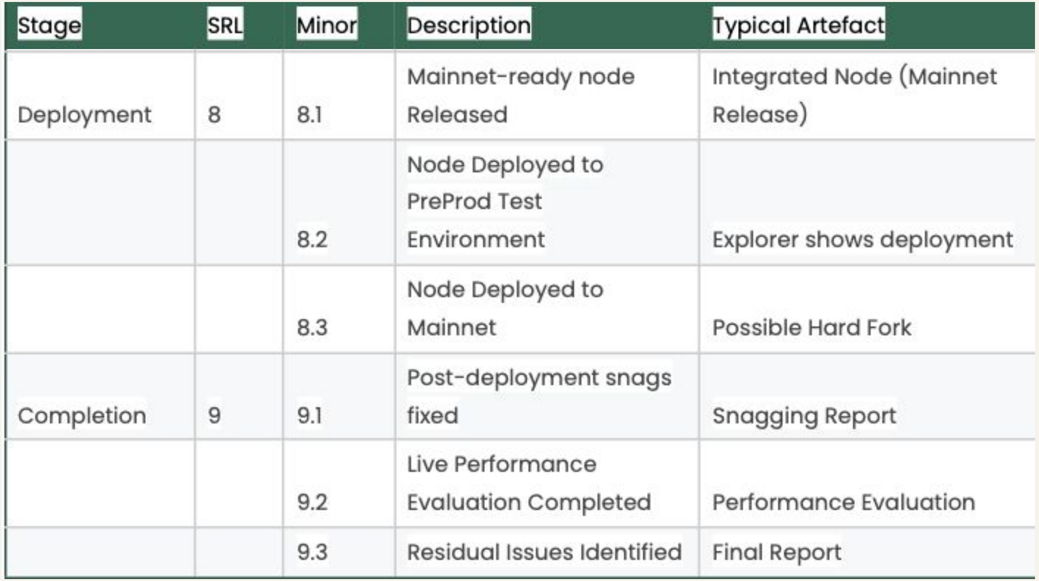


Stage	SRL	Minor	Description	Typical Artefact
Main Implementation	6	6.1	Code Completed	Software Deliverable
		6.2	Functionality Tests Completed	Test Results
		6.3	Security Review Completed	Security Audit Reports
		6.4	Performance Tests Completed as Necessary	Performance Reports
Integration	7	7.1	Integration Into Node	Integrated Node (Preview Release)
		7.2	Integration Functionality Tests Completed	Test Results
		7.3	Node Deployed to Preview Test Environment	Explorer shows deployment
		7.4	Performance Tests Completed	Performance Results



INTERSECT™

Software Readiness Levels (SRLs): 6-7



Stage	SRL	Minor	Description	Typical Artefact
Deployment	8	8.1	Mainnet-ready node Released	Integrated Node (Mainnet Release)
		8.2	Node Deployed to PreProd Test Environment	Explorer shows deployment
		8.3	Node Deployed to Mainnet	Possible Hard Fork
Completion	9	9.1	Post-deployment snags fixed	Snagging Report
		9.2	Live Performance Evaluation Completed	Performance Evaluation
		9.3	Residual Issues Identified	Final Report