



Van Innoveren naar Engineeren zonder Frustreren

Raakvlakken, informatie mapping en een werkbare overdracht

Managementsamenvatting

Rapid Learning Cycles (RLC) en Systems Engineering (SE) zijn complementair, niet concurrerend. RLC is een discipline voor kennisopbouw in de fuzzy front end, waar de kosten van een verkeerde beslissing hoog zijn en gevalideerde kennis nog schaars is. SE is een gestructureerde discipline die een stabiel concept omzet in een geverifieerd, gevalideerd en geïntegreerd systeem. De twee werken het best na elkaar, met een bewuste en goed ingerichte overdracht: **RLC sluit kennishiaten en stabiliseert sleutelbeslissingen; SE erft een stabiele basis en vertaalt die naar requirements, architectuur, ontwerp, verificatie, validatie en uiteindelijk een geleverd systeem.**

Deze briefing beschrijft de tien raakvlakken waarop RLC- en SE-artefacten aan elkaar worden overgedragen, de gate waarop die overdracht moet plaatsvinden en de operationele risico's die het engineeringmanagement moet beheersen. Het doel is een naadloze kennisstroom – zonder opnieuw baselinen en zonder het vermogen te verliezen om beslissingen te herzien wanneer nieuw bewijs beschikbaar komt.

Belangrijkste inzichten

- RLC gaat vooraf aan SE, maar de grens is een beheerde overgang, geen muur. De Core Hypothesis, het Key Decisions-logboek en de Knowledge Repository voeden rechtstreeks de SE-werkproducten.
- Tien raakvlakken op artefactniveau dragen kennis van RLC over naar SE; elk heeft een gedefinieerde producent, consument en gate.
- De Concept Stability Gate is het centrale moment waarop het management de overgang autoriseert. De gate wordt gestuurd op het sluitingspercentage van sleutelbeslissingen en het restrisico van kennishiaten, niet op de kalender.
- Behandel SE-requirements als erfgenamen van RLC-beslissingen, niet als een parallel beslislogboek. Dubbele beslissingsbevoegdheid is de meest voorkomende faalmodus.
- Reserveer een pad voor herintrede in RLC. Wanneer SE-verificatie een aanname ontkracht, is de reactie een afgebakende leercyclus, geen herontwerp.

1. De twee raamwerken in één oogopslag

Rapid Learning Cycles (RLC)

RLC, geformaliseerd door Katherine Radeka, is een op cadans gebaseerd raamwerk voor de voorkant van productontwikkeling. Het team identificeert **sleutelbeslissingen** – beslissingen die het ontwerp sturen en niet eenvoudig terug te draaien zijn – brengt de **kennishiaten** in kaart die deze beslissingen blokkeren en voert korte leercycli uit, meestal van twee tot zes weken, om de hiaten te sluiten. Voortgang wordt gemeten in afgesloten kennis, niet in opgeleverde deliverables. Het raamwerk convergeert naar een **stabiel concept**: een Core Hypothesis, een set opgeloste sleutelbeslissingen en een Knowledge Repository waarop de rest van de organisatie met vertrouwen kan voortbouwen.

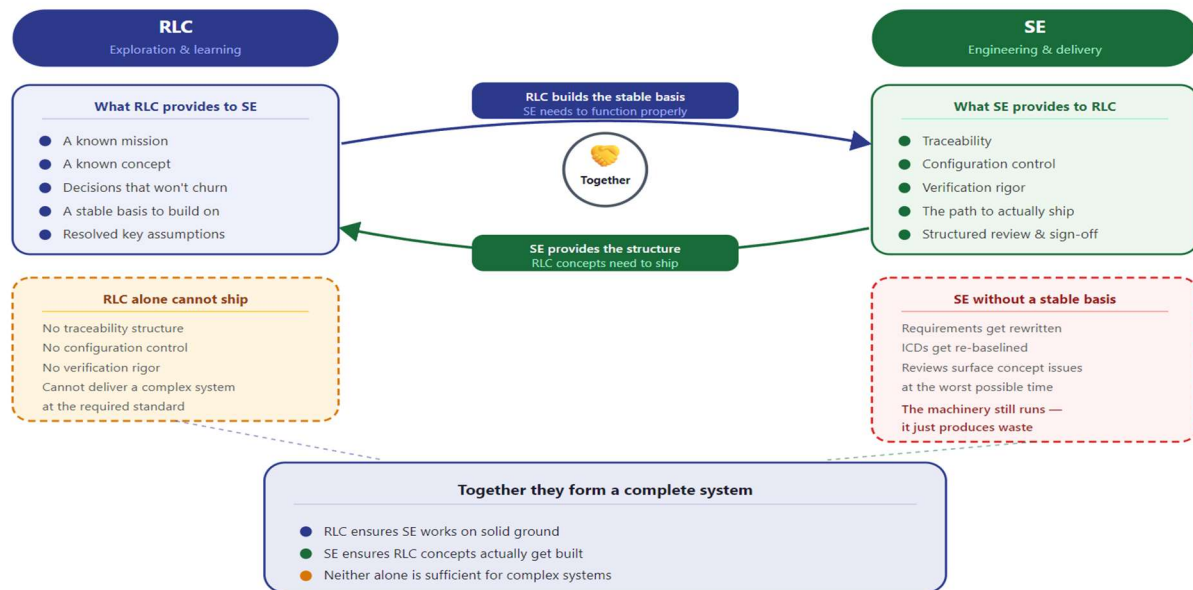
Systems Engineering (SE)

SE, zoals vastgelegd door INCOSE en de ISO/IEC/IEEE 15288-levenscyclus, is een gestructureerde discipline voor het ontwikkelen van complexe systemen. Langs het V-model haalt SE stakeholderbehoeften op, definieert systeem- en subsysteemrequirements, ontwikkelt architectuur en interfaces, alloceert ontwerp naar subsystemen en verifieert en valideert vervolgens het geïntegreerde systeem. SE steunt op traceerbare artefacten – requirements, ICD's, N²-diagrammen, trade studies, TPM's en risicoregisters – en formele technische reviews zoals SRR, PDR en CDR om de kosten van verandering te beheersen naarmate het ontwerp volwassen wordt.

Waarom ze elkaar nodig hebben

RLC en SE lossen elk een probleem op dat de ander niet kan oplossen. RLC is ontworpen voor onzekerheid – het is het meest waardevol wanneer je nog niet weet of je het juiste bouwt, en wanneer het vastleggen van een verkeerde beslissing kostbaar is. SE is ontworpen voor uitvoering – het neemt een stabiel concept en zet dat om in een geverifieerd, geleverd systeem. Op zichzelf is geen van beide voldoende: RLC kan blijven verkennen en leren, maar kan geen complex systeem opleveren; SE kan rigoreus engineeren, maar veroorzaakt verspilling als het onderliggende concept nog verschuift.

De combinatie werkt omdat ze sequentieel zijn in plaats van concurrerend. RLC reduceert de onbekenden; SE erft het resultaat en bouwt daarop voort. Wanneer de overdracht goed wordt beheerd – via gedefinieerde raakvlakken en een formele gate – stroomt kennis door zonder verloren te gaan of te worden gedupliceerd.



Wanneer dit het belangrijkste is

Dit is vooral belangrijk op twee specifieke momenten:

1. Het eerste moment is aan de voorkant, wanneer het concept nog vaag is. Daar zijn de kosten van een slechte beslissing het hoogst en is gevalideerde kennis het laagst. SE hier te vroeg inzetten — voordat het concept stabiel is — leidt tot requirements die opnieuw moeten worden geschreven en reviews die conceptuele problemen blootleggen op het slechtst mogelijke moment.
2. Het tweede moment is wanneer SE-verificatie aantoont dat een eerdere aanname onjuist was. Op dat moment is de verleiding groot om informeel te herontwerpen. De gecombineerde aanpak biedt juist een gestructureerd pad terug: een afgebakende leercyclus, geen ad-hocoplossing. Dit houdt de beslissingsbevoegdheid helder en voorkomt de meest voorkomende faalmodus: parallelle besluitvorming door RLC- en SE-teams.

2. Het sequentiemodel

RLC komt vóór SE, maar de overgang is geen eenvoudige overdracht. Het is een beheerde progressie met bewuste overlap en een gedefinieerd pad voor herintrede.

Toestand A — RLC-geleid (conceptverkenning)

Het verkenningsteam heeft de leiding. Zij bepalen het ritme, nemen de sleutelbeslissingen en volgen wat nog onbekend is. Engineers zijn betrokken, maar vooral als observatoren: zij nemen deel aan overleggen om vast te leggen wat ontstaat, zoals grove architectuur, interfaces en risico's, zodat zij klaarstaan wanneer hun rol start. Nog niets ligt vast; documenten die engineers in deze fase maken zijn vroege concepten, geen officiële commitments.

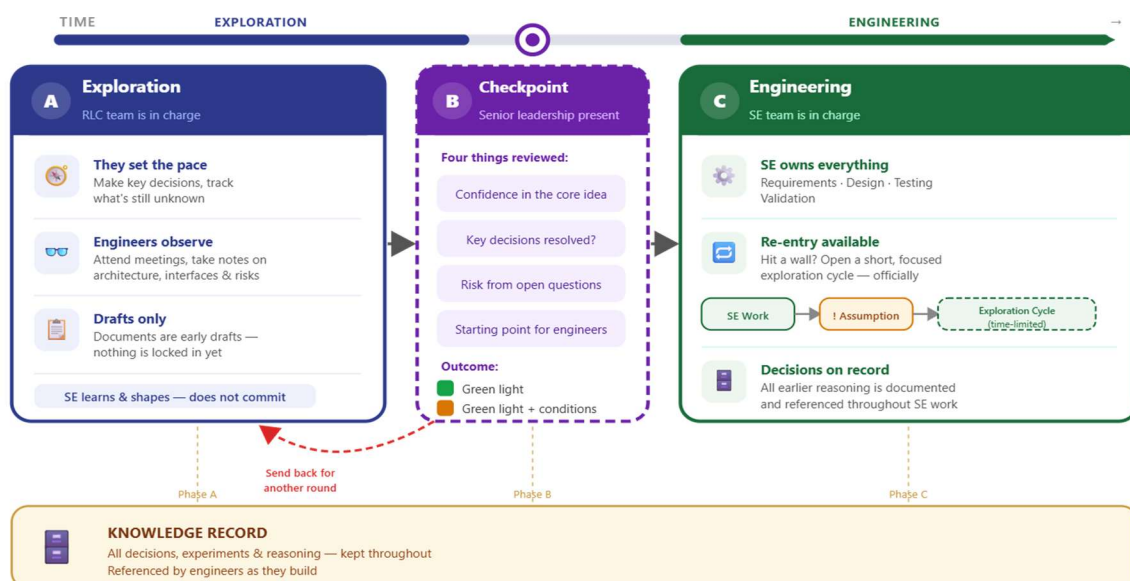
Toestand B – Concept Stability Gate (de overdracht)

Voordat engineering het overneemt, vindt een formeel checkpoint met senior management plaats. Vier zaken worden beoordeeld: hoe zeker het team is van het kernidee, hoeveel sleutelbeslissingen daadwerkelijk zijn opgelost, hoeveel risico resteert door onbeantwoorde vragen en hoe het startpunt voor engineering eruitziet.

De uitkomst is één van drie opties: groen licht geven, groen licht geven met voorwaarden – waarbij enkele open vragen mogen worden meegenomen mits ze worden gevolgd – of het verkenningsteam terugsturen voor een extra ronde.

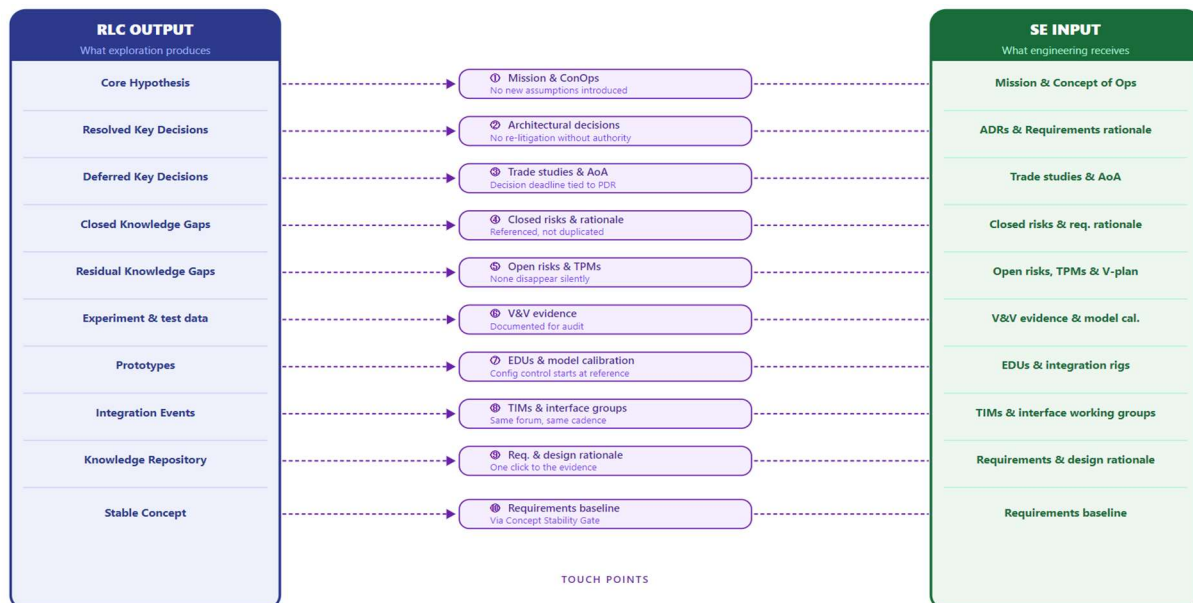
Toestand C – SE-geleid (met mogelijkheid tot RLC-herintrede)

Het engineeringteam is nu eigenaar van requirements, ontwerp, testen en validatie. Maar de verkenningfase verdwijnt niet. Als engineers vastlopen – bijvoorbeeld wanneer een prestatiedoel onhaalbaar blijkt of testresultaten een eerdere aanname tegenspreken – kan het team een korte, gerichte verkenningscyclus openen om dit uit te zoeken. Dit is een officieel, tijdgebonden proces, geen informeel herontwerpoverleg. Alle redeneringen achter eerdere beslissingen blijven vastgelegd en worden geraadpleegd terwijl engineers hun werk doen.



3. De tien raakvlakken

Elke rij identificeert een RLC-artefact, het SE-artefact dat ermee wordt gevoed, de overdrachtsgate waarop de overdracht wordt erkend en de eigenaar die verantwoordelijk is voor de overdracht. De rijen zijn geordend volgens de natuurlijke informatiestroom van concept naar verificatie.



4. Raakvlakken in detail

Elk raakvlak wordt beschreven als een producerend artefact, consumerend artefact en de toets voor een schone overdracht. Die toets bepaalt wat het management bij de Concept Stability Gate moet verifiëren.

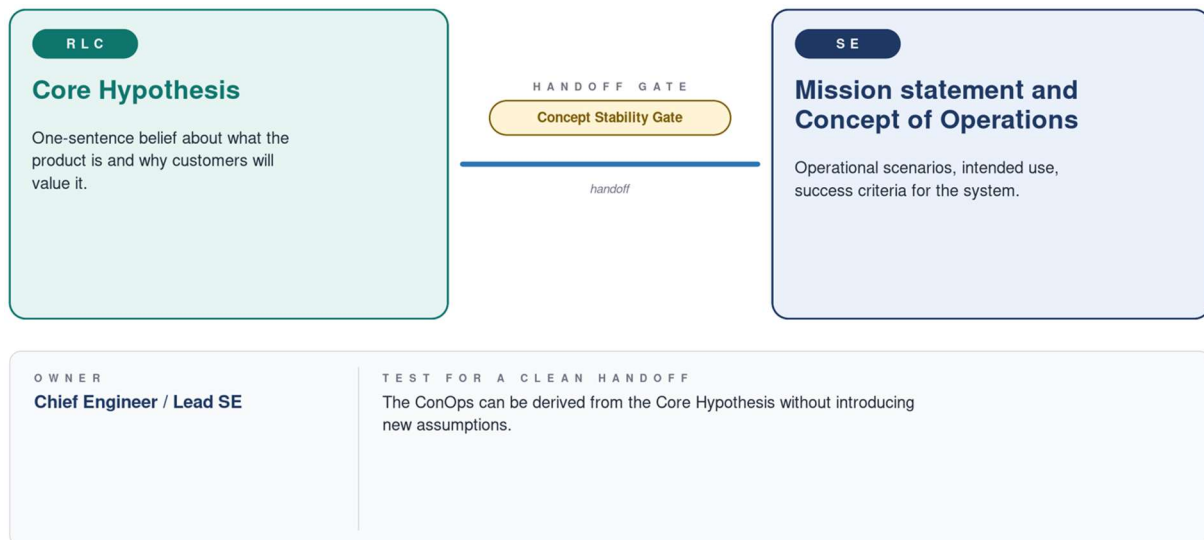
Raakvlak 1 – Core Hypothesis → Missie & Concept of Operations

De Core Hypothesis is de eenduidige RLC-beschrijving van wat het product is en waarom klanten er waarde aan hechten. In SE wordt hetzelfde beeld vastgelegd in de missieverklaring en het Concept of Operations.

Toets voor een schone overdracht: de ConOps moet logisch volgen uit de Core Hypothesis zonder nieuwe aannames toe te voegen, en elk operationeel scenario moet traceerbaar zijn naar bewijs in de Knowledge Repository of naar een benoemd en geaccepteerd open kennishiaat.

Gereed wanneer: elk operationeel scenario traceert naar een entry in de Knowledge Repository of een benoemd, geaccepteerd open kennishiaat; de missieverklaring is goedgekeurd door de Chief Engineer; en de ConOps geen verborgen aannames bevat.

Veelvoorkomende valkuil: de ConOps kopiëren uit een marketingbrief of een eerder projectplan zonder deze te valideren tegen de Knowledge Repository. Elk scenario dat niet op bewijs is gebaseerd, is een onbeproefde aanname die later – tegen hogere kosten – tijdens SE-verificatie naar boven komt.



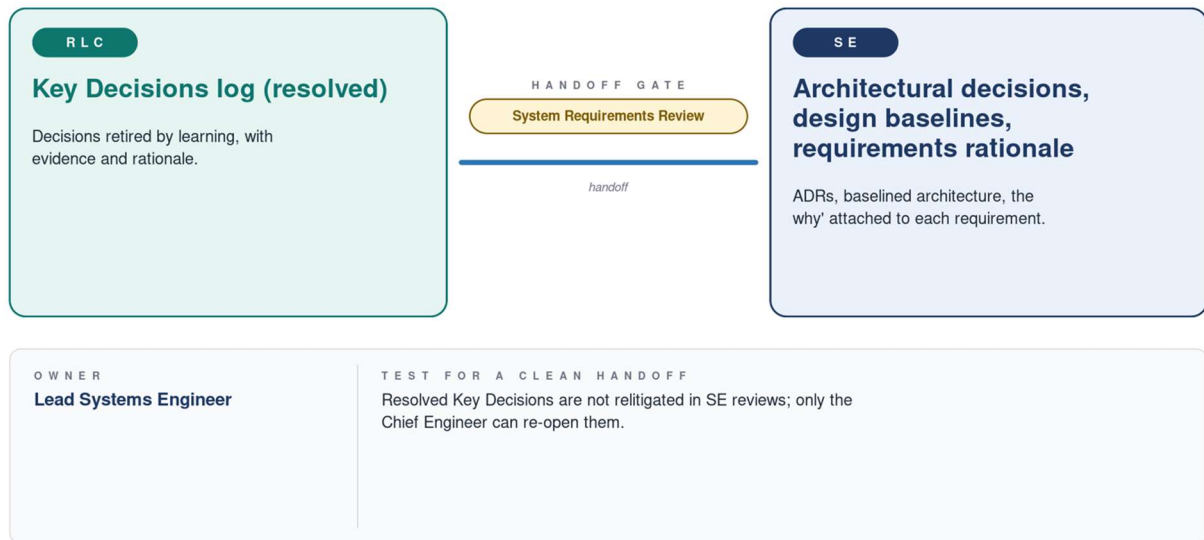
Raakvlak 2 – Opgeloste sleutelbeslissingen → Architectuurbeslissingen & requirements-rationale

Elke sleutelbeslissing die RLC heeft opgelost, wordt óf een architectuurbeslissing – vastgelegd in een ADR of architectuurdocument – óf een constraint die doorwerkt in requirements.

Discipline: heropen opgeloste sleutelbeslissingen niet tijdens SE. Het opnieuw openen vereist dezelfde bevoegdheid als het sluiten ervan, meestal de Chief Engineer, en een gecharterde leercyclus – geen zijgesprek in een design review.

Gereed wanneer: elke gesloten sleutelbeslissing traceerbaar is naar ten minste één ADR of requirements-rationale, en het SE-team elke architectuurkeuze kan uitleggen zonder opnieuw het RLC-team te raadplegen.

Veelvoorkomende valkuil: toestaan dat SE-designreviews opgeloste sleutelbeslissingen informeel opnieuw ter discussie stellen. Zodra een sleutelbeslissing is gesloten, vereist heropening bevoegdheid van de Chief Engineer en een gecharterde leercyclus – geen zijgesprek in een PDR-voorbereiding.



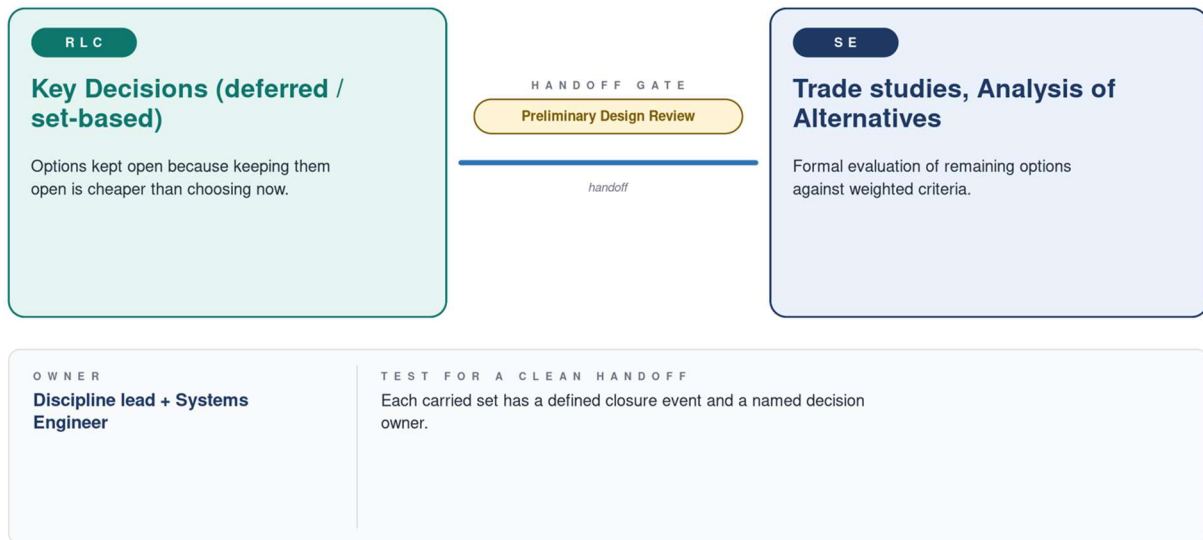
Raakvlak 3 – Uitgestelde / set-based sleutelbeslissingen → Trade studies & Analyse van Alternatieven

RLC staat expliciet toe dat beslissingen worden uitgesteld wanneer het goedkoper is om opties open te houden dan vroegtijdig te kiezen. Deze meegenomen optiesets worden in SE vertaald naar trade studies of Analyses van Alternatieven, met een vastgelegde beslisdeadline die is gekoppeld aan een downstream SE-event, meestal PDR.

Te beheersen risico: set-based opties hebben kosten – houd alleen de opties in stand die overeind blijven nadat de afsluitende kennishiaten zijn getoetst.

Gereed wanneer: elke uitgestelde sleutelbeslissing een eigenaar, een harde afsluitdeadline en een trade study-charter heeft; geen enkele optie actief wordt doorontwikkeld nadat de bovenliggende beslissing is gesloten; en alle PDR-kritische beslissingen vóór de PDR-gate zijn opgelost.

Veelvoorkomende valkuil: opties voor de zekerheid openhouden terwijl de data al duidelijk naar één keuze wijzen. Een dode optie in leven houden verbruikt engineeringcapaciteit en dwingt interfaceteams om onnodig ontwerpruimte open te houden.

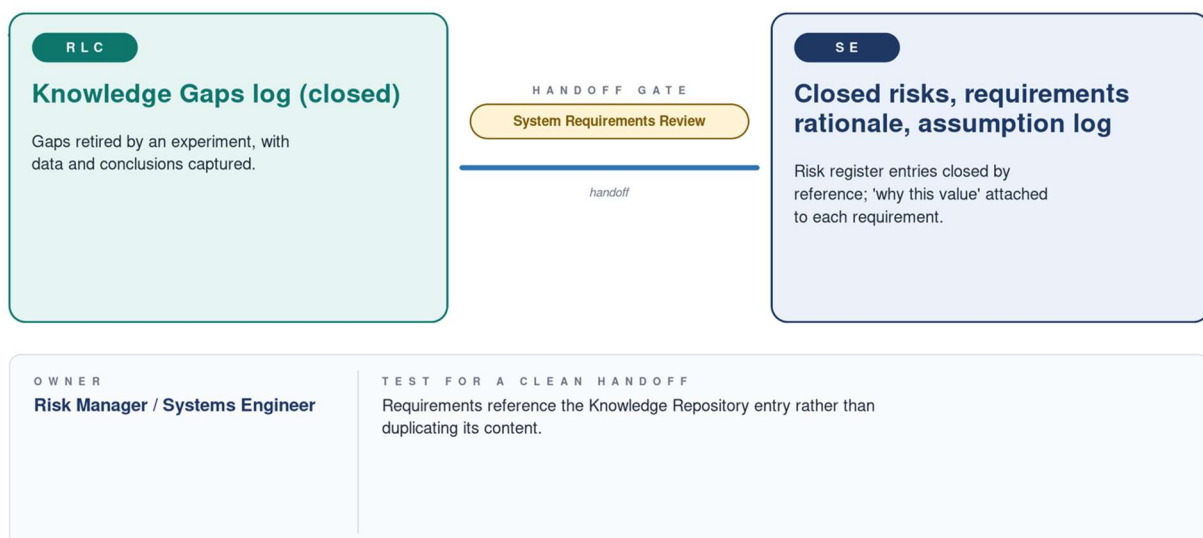


Raakvlak 4 – Gesloten kennishiaten → Gesloten risico's & requirements-rationale

Een kennishiaat dat door een experiment is gesloten, wordt óf een gesloten risico in het SE-risicoregister óf, vaker, de rationale die aan een requirement wordt gekoppeld ("waarom deze waarde, waarom deze tolerantie"). Naar de entry in de Knowledge Repository wordt verwezen; deze wordt niet gedupliceerd in het requirementsdocument.

Gereed wanneer: er geen gesloten kennishiaat bestaat zonder bijbehorende gesloten entry in het risicoregister of annotatie bij de requirements-rationale; en elke requirement met een niet-triviale waarde traceerbaar is naar een experiment.

Veelvoorkomende valkuil: een kennishiaat als gesloten beschouwen op administratieve gronden – een statusupdate zonder daadwerkelijk experiment of validatie. Verifieer altijd dat afsluiting wordt onderbouwd door bewijs, niet door optimisme.



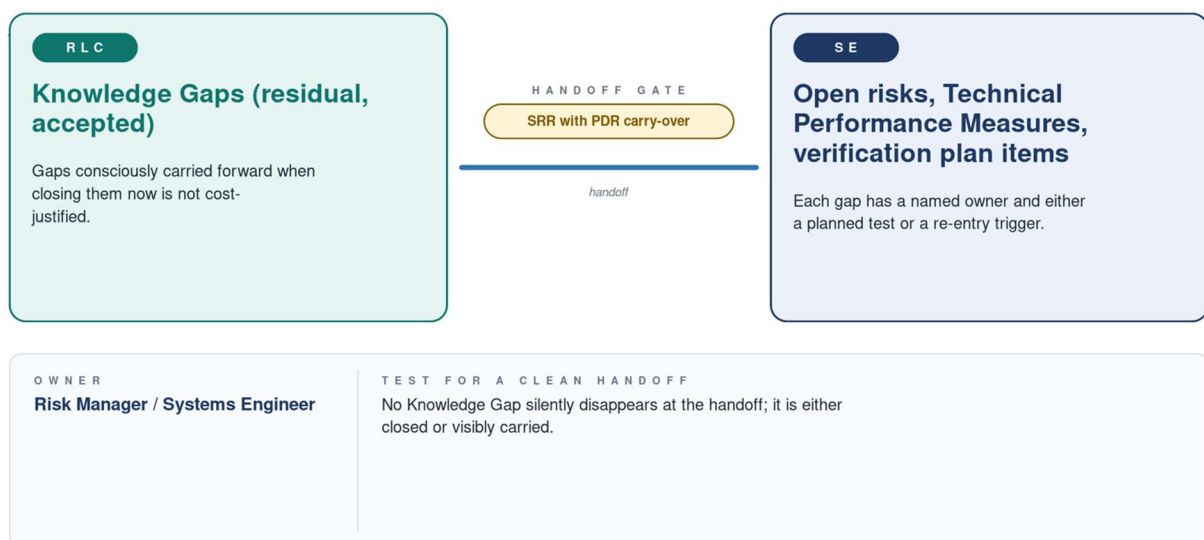
Raakvlak 5 — Resterende kennishiaten → Open risico's, TPM's, items in het verificatieplan

Niet elk kennishiaat kan worden gesloten voordat SE begint. De hiaten die overblijven, worden omgezet in open SE-risico's met benoemde eigenaren, in TPM's die tijdens de ontwikkeling worden gevolgd, of in items in het verificatieplan waarmee het hiaat door testen wordt gesloten.

Managementregel: geen enkel kennishiaat verdwijnt stilzwijgend bij de overdracht. Het is óf gesloten, óf zichtbaar meegenomen.

Gereed wanneer: elk resterend kennishiaat voorkomt in het risicoregister, de TPM-lijst of het verificatieplan; elke vervolgactie een benoemde eigenaar, afsluitdeadline en acceptatiecriterium heeft; en geen enkel hiaat stilzwijgend is verdwenen bij de overdracht.

Veelvoorkomende valkuil: te veel hiaten classificeren als items in het verificatieplan, omdat testen minder verplichtend voelt dan mitigatie. Testen sluit onzekerheid laat en tegen hoge kosten — reserveer categorie C voor hiaten die werkelijk niet eerder kunnen worden aangepakt.



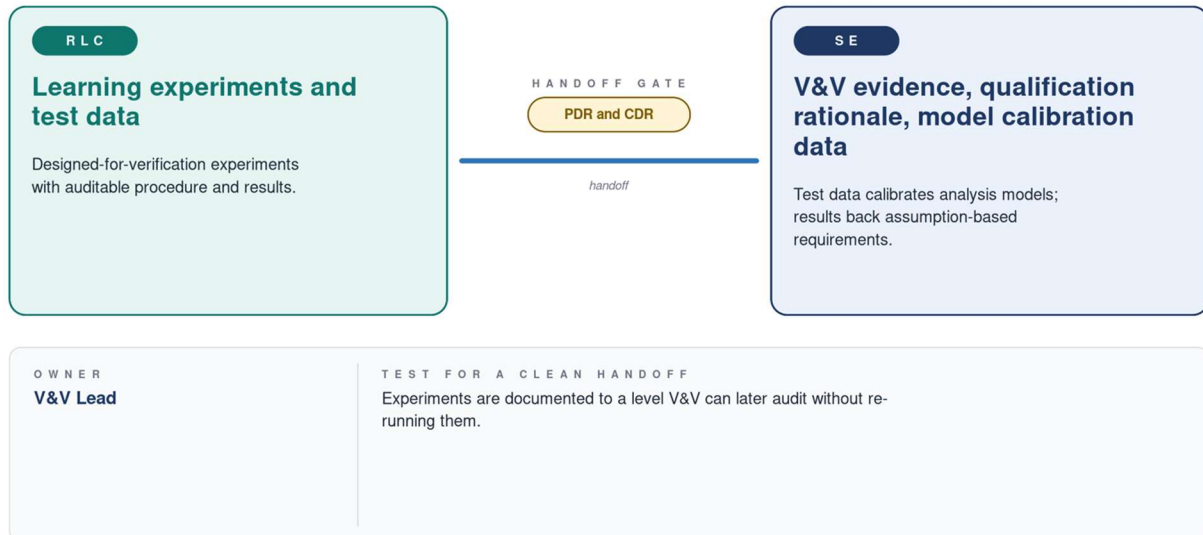
Raakvlak 6 — Leerexperimenten & testdata → V&V-bewijs

RLC-experimenten zijn geen wegwerpmateriaal. Wanneer ze met verificatie in gedachten worden ontworpen, kalibreren hun data de analysemodellen die SE gebruikt voor ontwerpverificatie en kwalificeren ze de aannames achter prestatie-eisen. Dit vraagt alleen dat RLC-experimenten worden gedocumenteerd op een niveau dat V&V later kan auditen — een kleine extra inspanning met aanzienlijke downstream opbrengst.

Gereed wanneer: elke requirement met een prestatiewaarde óf een experimentele referentie heeft óf een open item in het verificatieplan; elk analysemodel dat in SE wordt gebruikt een gedocumenteerde kalibratiebasis heeft; en de V&V Lead de oorsprong van elk bewijsitem kan traceren.

Veelvoorkomende valkuil: aannemen dat een positief resultaat van een RLC-experiment automatisch kwalificatiebewijs vormt. Kwalificatie vereist specifieke condities,

traceerbaarheid naar een toepasselijke norm en vaak een formeel bijgewoonde test. Identificeer het verschil vroeg en plan dienovereenkomstig.

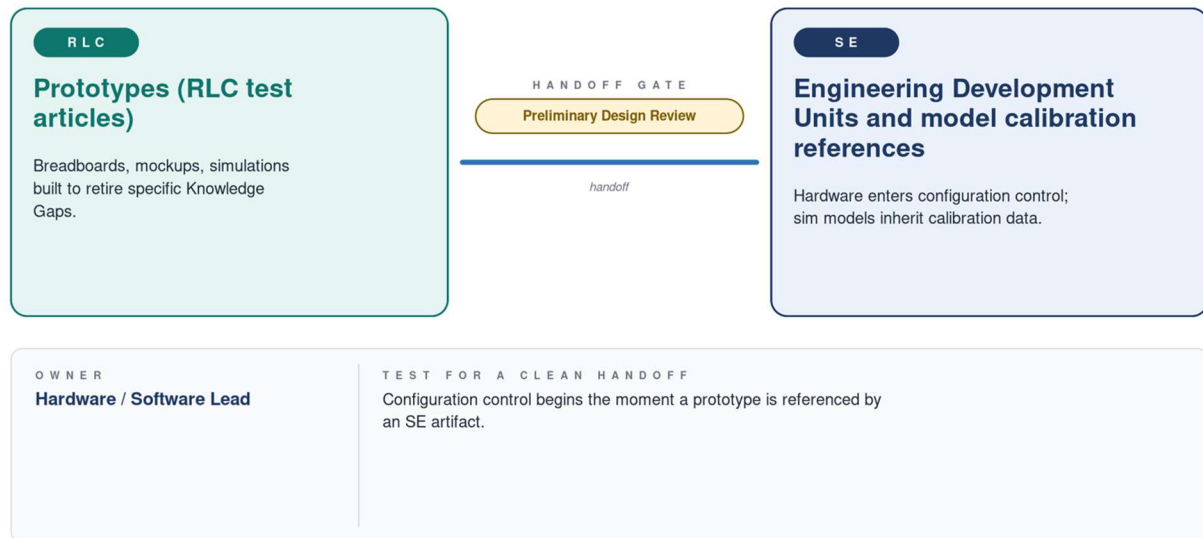


Raakvlak 7 – Prototypes → Engineering Development Units & modelkalibratie

RLC-prototypes – breadboards, mock-ups, simulaties – blijven vaak bestaan nadat hun leerdoel is bereikt. SE erft ze als EDU's, als kalibratiereferenties voor modellen met hogere nauwkeurigheid, of als basis voor vroege integratie-opstellingen. Configuratiebeheer begint op het moment dat een prototype door een SE-artefact wordt genoemd – niet eerder.

Gereed wanneer: elk prototype met SE-bestemming A, B of C onder configuratiebeheer staat met een beperkingenkaart; de Hardware/Software Lead precies weet wat elk prototype wel en niet kan aantonen; en geen enkel prototype in SE-testen wordt gebruikt zonder dat de beperkingen zijn gedocumenteerd.

Veelvoorkomende valkuil: een prototype overdragen zonder beperkingenkaart. De SE-engineer zal dan aannemen dat het representatiever is dan het werkelijk is, wat leidt tot te zelfverzekerde conclusies uit integratietests die zijn gebaseerd op een onvolledig of vereenvoudigd object.

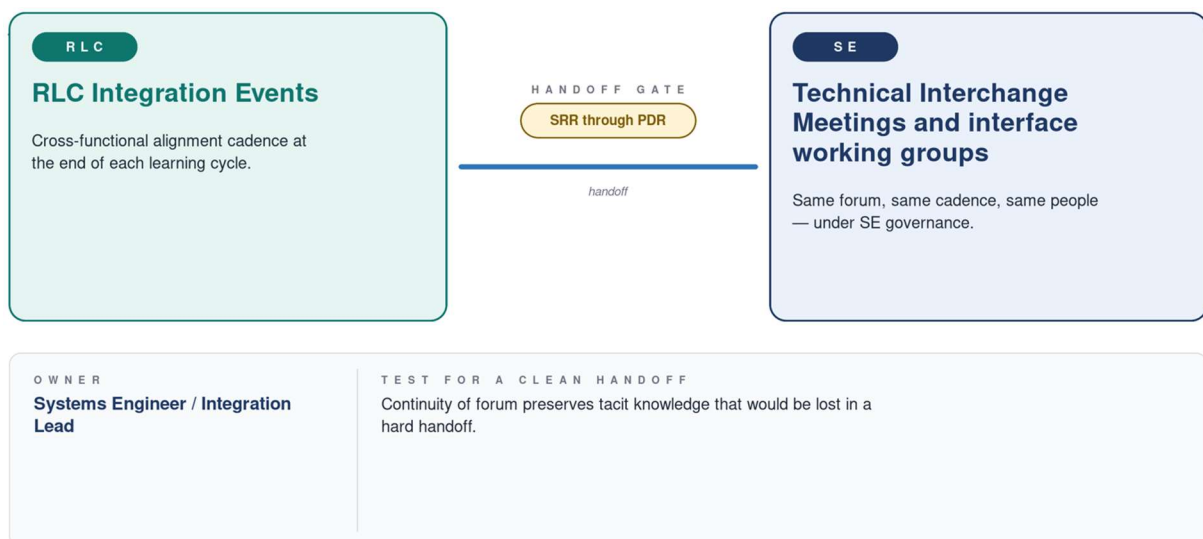


Raakvlak 8 – Integratie-events → TIM's en interfacewerkgroepen

RLC-integratie-events brengen het crossfunctionele team samen om af te stemmen op nieuwe kennis. Wanneer het programma overgaat, wordt hetzelfde forum, met dezelfde cadans en veel van dezelfde deelnemers, de SE Technical Interchange Meeting of interfacewerkgroep. Continuïteit van het forum behoudt de sociale en technische kennis die anders bij een harde overdracht verloren zou gaan.

Gereed wanneer: de TIM het erkende centrale forum is voor interfacestatus en open actiepunten; elke kritische interface een IWG heeft met een actieve eigenaar en vergadercadans; en TIM-verslagen dienen als kennisbron waardoor het minder nodig is om het RLC-team rechtstreeks te raadplegen.

Veelvoorkomende valkuil: een TIM opzetten met alleen SE-engineers en daarmee de crossfunctionele breedte van het integratie-event verliezen. Juist die crossfunctionele samenstelling maakte integratie-events effectief – behoud die.

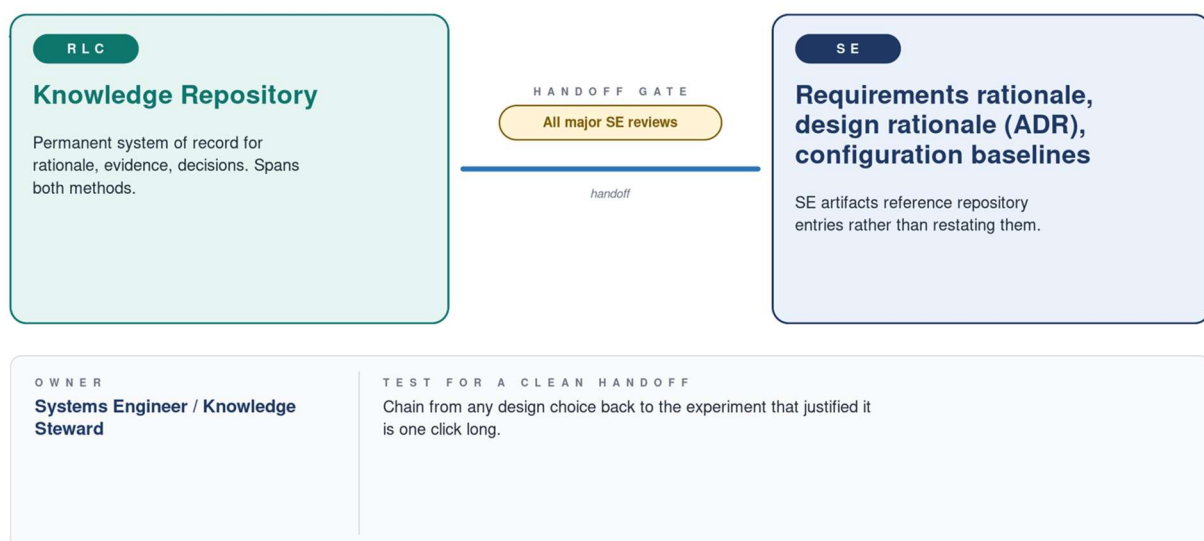


Raakvlak 9 – Knowledge Repository → Requirements- & ontwerprationale

De Knowledge Repository is het enige artefact dat beide methoden overspant. Behandel deze als een permanent systeem van vastlegging. Requirements-rationale, ontwerprationale en trade study-rapporten moeten verwijzen naar repository-entries in plaats van ze te herhalen. Dit maakt de gecombineerde aanpak traceerbaar: wanneer een ontwerpkeuze bij CDR ter discussie wordt gesteld, is de keten terug naar het experiment dat deze keuze onderbouwde direct beschikbaar.

Gereed wanneer: elke requirement met een niet-triviale waarde een referentie naar de Knowledge Repository heeft of een expliciet gedocumenteerde uitzondering; elke ADR naar de repository verwijst; en de traceerbaarheidsketen van elke betwiste ontwerpbeslissing terug naar het oorspronkelijke experiment bij CDR kan worden aangetoond.

Veelvoorkomende valkuil: de Knowledge Repository bevroren bij de Concept Stability Gate en behandelen als een gesloten archief. Nieuwe SE-inzichten die eerdere redeneringen tegenspreken of verfijnen, moeten worden vastgelegd – anders misleidt de repository in plaats van te informeren.

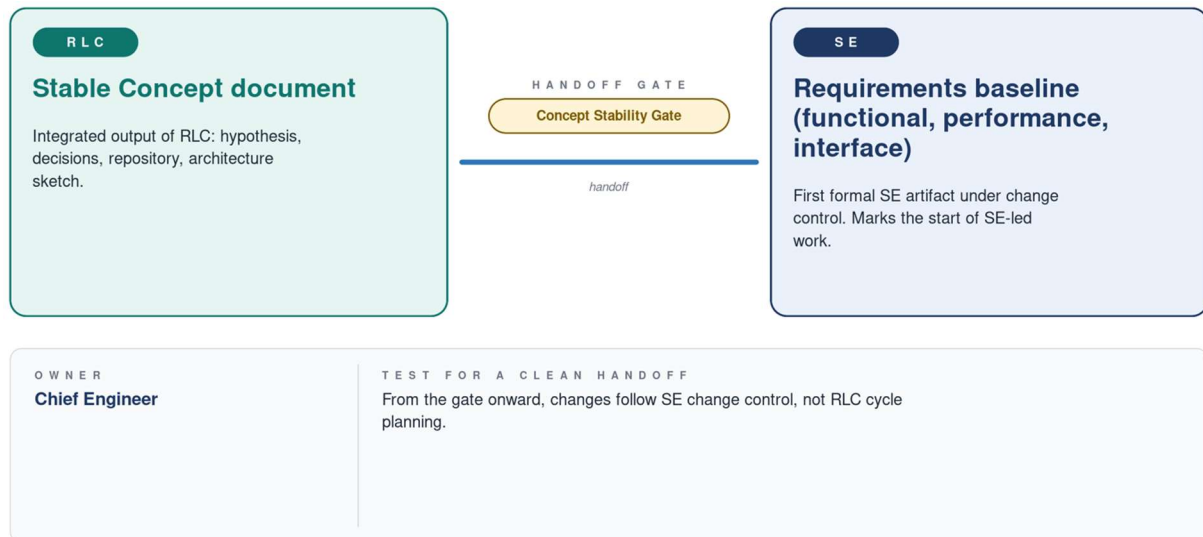


Raakvlak 10 – Stabiel concept → Requirementsbaseline

Het stabiele concept is de geïntegreerde output van RLC – een Core Hypothesis, een Key Decisions-logboek, een Knowledge Repository en de opgeloste architectuurschets. De eerste taak van SE is dit te vertalen naar een formele requirementsbaseline. De Concept Stability Gate is het moment waarop het management het stabiele concept accepteert als basis voor SE-werk; vanaf dat moment volgen wijzigingen het SE-wijzigingsbeheerproces.

Gereed wanneer: elke requirement een rationale heeft of als TBR is gelabeld met een eigenaar en oplossingsplan; de set requirements intern consistent is; de Chief Engineer de baseline formeel heeft geaccepteerd; en elke wijziging na de baseline het SE-wijzigingsbeheerproces volgt.

Veelvoorkomende valkuil: requirements schrijven vanuit de oplossing in plaats van vanuit de stakeholderbehoefte. Oplossingsgedreven requirements beperken het ontwerp te sterk, leggen implementatiekeuzes vast voordat de ontwerpruimte is verkend en leveren een baseline op die moeilijker te onderhouden is naarmate het ontwerp evolueert.



5. De Concept Stability Gate (checkpoint)

De Concept Stability Gate is het enige beslismoment waarop het management de overgang van RLC-geleid naar SE-geleid werk autoriseert. Het is het enige governance-mechanisme dat nodig is om de twee methoden te combineren. De gate heeft vijf criteria; slagen is binair, eventueel met aanvullende voorwaarden.

5 CRITERIA — ALL MUST BE SATISFIED			EVIDENCE SOURCE
	CRITERION	WHAT "GOOD" LOOKS LIKE	
① CORE HYP.	Core Hypothesis confidence	Team defends with evidence, not opinion. Customers have validated the value proposition.	<i>Knowledge Repository Customer learning experiments</i>
② KEY DECISIONS	Key Decision closure	Critical path decisions closed. Open ones are set-based with a named closure event.	<i>Key Decisions log</i>
③ KG RISK	Residual Knowledge Gap risk	Each gap: named owner, accepted risk posture, and verification or re-entry trigger.	<i>Knowledge Gaps log Draft risk register</i>
④ SE BASELINE	SE entry baseline readiness	Draft ConOps, req. themes, architecture sketch coherent with the Stable Concept.	<i>SE stubs developed during State A</i>
⑤ ORG. READY	Organizational readiness	SE assigned, V&V Lead engaged, configuration control plan defined.	<i>Program plan</i>

DECISION OPTIONS

✓ PASS

Authorize transition.
SE takes the lead at the next program day. RLC phase formally closes.
*All 5 criteria met — no conditions attached.
Clean handoff.*

— PASS WITH CONDITIONS

Authorize transition with named gaps.
Open Knowledge Gaps carried into the SE risk register.
A re-entry trigger defined for each gap.
*Conditions are tracked — not forgotten.
Each has a named owner.*

✗ HOLD

Extend RLC for a bounded cycle.
Typically one cycle — never more than two.
A second hold escalates to portfolio level.

Beslisopties bij de gate

- Pass — autoriseer de overgang. SE neemt de leiding op de eerstvolgende programmadag.
- Pass met voorwaarden — autoriseer de overgang, met benoemde kennisdelen die worden meegenomen in het SE-risicoregister en een herintredetrigger voor elk hiaat.
- Hold — verleng RLC met een afgebakende cyclus, meestal één cyclus en nooit meer dan twee. Een tweede hold moet escaleren naar een portfolio-overleg.

6. Risico's en antipatronen om te beheersen

Dubbele beslissingsbevoegdheid

Dit is de meest voorkomende faalmodus. Als zowel RLC-sleutelbeslissingen als SE-designreviews dezelfde vraag kunnen beslissen, zal het team uiteindelijk elke beslissing opnieuw ter discussie stellen. Mitigatie: zodra een sleutelbeslissing is gesloten, kan alleen de Chief Engineer deze heropenen, en uitsluitend door een leercyclus te charteren.

Voortijdige overgang

Managementdruk om “met engineering te starten” voordat het concept stabiel is, leidt tot requirementschurn en herwerk. De Concept Stability Gate bestaat om het management een verdedigbare taal te geven om deze druk te weerstaan – en om de trade-off zichtbaar te maken als zij ervoor kiezen deze toch te overrulen.

Verloren kennis bij de overdracht

Impliciete kennis bij het RLC-team overleeft zelden een harde overdracht. Mitigatie: blijvende overlegfora (raakvlak 8), een Systems Engineer die tijdens Toestand A is ingebed, en strikt gebruik van de Knowledge Repository (raakvlak 9).

Theatrale kennishiaten

Dit zijn hiaten die gemakkelijk te sluiten zijn, maar feitelijk geen sleutelbeslissing blokkeren. Ze verbruiken cycluscapaciteit zonder de besluitvorming echt vooruit te helpen. Mitigatie: elk kennishiaat traceert naar een sleutelbeslissing; hiaten die dat niet doen, worden uitgesteld of geschrapt.

RLC als permanente toestand

Teams geven soms de voorkeur aan de leerfase omdat commitments daar lichter voelen. Symptomen zijn afnemende cyclus-ROI en kennishiaten die versplinteren in plaats van sluiten. Mitigatie: een gepubliceerde gatekalender en management dat bereid is de gate daadwerkelijk te laten plaatsvinden.

Geen pad terug

Zodra SE begint, behandelen teams het heropenen van een sleutelbeslissing soms als falen. Dat is het niet. Mitigatie: charter een helder, afgebakend RLC-herintredepad met een expliciet afsluitmoment, zodat het team dit ziet als een normaal mechanisme en niet als een noodsituatie.

7. Aanbevelingen voor engineeringmanagement

- Neem de Concept Stability Gate op als formele gate in de programmalifecycle, onder eigenaarschap van de Chief Engineer en met formele managementgoedkeuring.
- Wijs vanaf dag één een Systems Engineer toe aan elk RLC-programma, in een consumerende rol, om raakvlakken 1, 8 en 9 te voeden.
- Definieer een template voor een RLC-herintredecharter, zodat de weg terug een bekend mechanisme is en geen uitzondering.
- Volg twee leading indicators over beide fasen heen: het sluitingspercentage van sleutelbeslissingen aan RLC-zijde en de requirementschurn aan SE-zijde. Aanhoudend hoge churn na de gate wijst op een voortijdige overgang.
- Train de SE-community in RLC-artefacten – hoe ze eruitzien en wat ze impliceren – en de RLC-community in SE-artefacten – wat zij moeten aanleveren. De vertaling is eenvoudig zodra de woordenschat gedeeld is.

Integratieraakvlakken, artefactmapping en een werkbaar overdrachtsmodel