

SDSE ROI and Business Cases

A Practical Guide to Funding Better Software Decisions

The central proposition

SDSE earns its return when a modest investment in behavioral understanding prevents a larger investment in building, repairing, operating, or defending the wrong behavior.

By Marc Hornbeek, a.k.a. DevOps_the_Gray

Originator of Simulation-Driven Software Engineering

August 2026

Executive Summary

Software leaders are under pressure to deliver more capability, faster, with fewer people and less risk. AI-assisted development increases the speed at which teams can turn an idea into working software. It does not guarantee that the idea was interpreted correctly, that interacting behaviors are safe, or that the resulting system will produce the business outcome stakeholders intended. Google Cloud's 2025 DORA research describes AI as an amplifier: it magnifies the strengths of capable delivery systems and the weaknesses of struggling ones [1].

Simulation-Driven Software Engineering (SDSE) addresses this behavioral assurance gap. Teams establish intent, create a Candidate Specification, express consequential behavior in an executable model, challenge that model with realistic scenarios, and approve a Validated Behavioral Baseline before expensive implementation and operational commitments harden. The living behavior model then supports design, code, testing, deployment, operations, and learning.

The financial argument is straightforward. NASA systems engineering guidance notes that life-cycle costs tend to become locked in early and that problems identified late cost considerably more to correct [2]. SDSE moves important behavioral discovery toward the part of the lifecycle where correction is still comparatively inexpensive. Its value can appear as avoided rework, faster decisions and delivery, reduced expected operational loss, recovered engineering capacity, and greater confidence in AI-enabled or otherwise consequential systems.

What this paper provides

A practical method for selecting an SDSE pilot, establishing a baseline, estimating costs and benefits, applying realization and risk factors, testing sensitivity, and presenting a decision-ready business case without unsupported promises.

1. Measure decisions and outcomes, not the number of models or simulations produced.
2. Separate cashable savings, recovered capacity, risk-adjusted avoided loss, and strategic value.
3. Use internal evidence, ranges, explicit assumptions, and a named owner for each estimate.
4. Scale only after a bounded pilot demonstrates useful value.

Contents

- 1. The Financial Problem Hiding in Software Delivery**
- 2. Why Conventional ROI Cases Miss Behavioral Risk**
- 3. How SDSE Creates Economic Value**
- 4. Five Sources of SDSE Business Value**
- 5. The SDSE ROI Model**
- 6. Risk-Adjusted Value and Sensitivity**
- 7. Illustrative Business Case: Autonomous Production Remediation**
- 8. Building a Business Case Leaders Will Accept**
- 9. Pilot Design and Measurement**
- 10. Measures That Matter**
- 11. What Different Leaders Need to Hear**
- 12. Common Mistakes and Objections**
- 13. Decision Framework**

Conclusion

Appendix A. SDSE Business Case Worksheet

References

About the Author

1. The Financial Problem Hiding in Software Delivery

Most software investment cases begin with development cost, delivery schedule, infrastructure, productivity, and expected revenue. They often treat incorrect behavior as a testing issue that will be controlled by normal quality practices. That assumption is increasingly fragile. Modern systems combine services, data, policies, interfaces, human workflows, external dependencies, and autonomous or AI-assisted decisions. Each local component may pass its tests while the combined behavior still violates the stakeholder promise.

The resulting cost rarely appears under one accounting label. It is distributed across product rework, architecture changes, additional testing, delayed releases, incident response, customer support, service credits, security remediation, compliance effort, management escalation, and lost opportunity. Because those costs are fragmented, the organization may repeatedly pay for behavioral ambiguity without recognizing it as a common economic problem.

Green is not proof

A green pipeline shows that configured checks passed. It does not prove that the complete system will behave correctly across state, sequence, interaction, failure, misuse, recovery, and changing operating conditions.

A practical example

Consider an AI-enabled production agent authorized to restart workloads, scale capacity, change traffic routing, and roll back releases. Each action may be permitted and individually tested. Yet a sequence of technically permitted actions can still worsen an outage. If the team discovers that behavior in a simulation, it changes a model and a control policy. If it discovers the same behavior in production, it may face customer impact, emergency labor, recovery cost, regulatory questions, and lost trust.

The ROI question is therefore not, "What is a behavior model worth?" The useful question is, "What decisions can SDSE improve, what expensive outcomes can those decisions change, and what evidence will show that the change occurred?"

2. Why Conventional ROI Cases Miss Behavioral Risk

Traditional business cases are strongest when cost and benefit can be traced to observable evidence. They become weak when optimistic productivity percentages are applied to broad populations or when prevented failures are treated as certain savings. A credible SDSE case must make the chain from engineering activity to business value explicit.

Discovery point	Typical response	Economic effect
Before implementation	Clarify intent, revise the Candidate Specification, model, or scenario.	Low direct correction cost; high leverage.
During development	Rewrite code, tests, prompts, policies, interfaces, and documentation.	Additional labor and schedule impact.
During integration or release	Coordinate changes across teams, repeat assurance, and delay deployment.	Cross-team rework and opportunity cost.
In production	Respond, recover, compensate, communicate, investigate, and redesign.	Operational loss, customer harm, and trust impact.

Table 1. Late discovery expands both the number of affected artifacts and the cost of correction.

NASA makes the same economic point at the systems level: major cost commitments are shaped early, while late identification and correction are more expensive [2]. SDSE applies that principle specifically to software behavior. It creates evidence before code, integration, and operational dependency make a behavioral correction costly.

Four disciplines for a credible estimate

The U.S. Government Accountability Office describes reliable cost estimates as comprehensive, well documented, accurate, and credible [3][4]. Those characteristics translate directly to an SDSE business case:

5. Comprehensive: include training, modeling, integration, review, ownership, and maintenance - not only tool cost.
6. Well documented: identify the source, owner, period, formula, and assumptions behind every material input.
7. Accurate: use current internal data and update the estimate as pilot evidence replaces assumptions.
8. Credible: present ranges, test sensitivity, avoid double counting, and invite independent challenge.

3. How SDSE Creates Economic Value

SDSE is a behavior-first engineering discipline. It begins with stakeholder intent and a Candidate Specification. The team models consequential behavior, executes scenarios, examines traces and counterexamples, revises assumptions, and approves a versioned Validated Behavioral Baseline. That baseline guides implementation and remains connected to operational evidence.

The economic mechanism is earlier, better-informed commitment. The organization spends a controlled amount to understand high-consequence behavior before it spends a larger amount to implement, integrate, release, operate, and repair that behavior. SDSE does not eliminate uncertainty. It changes when uncertainty becomes visible and how decision makers respond to it.

THE SDSE VALUE REALIZATION CHAIN

A credible business case connects engineering activity to observable change and then to financial value.

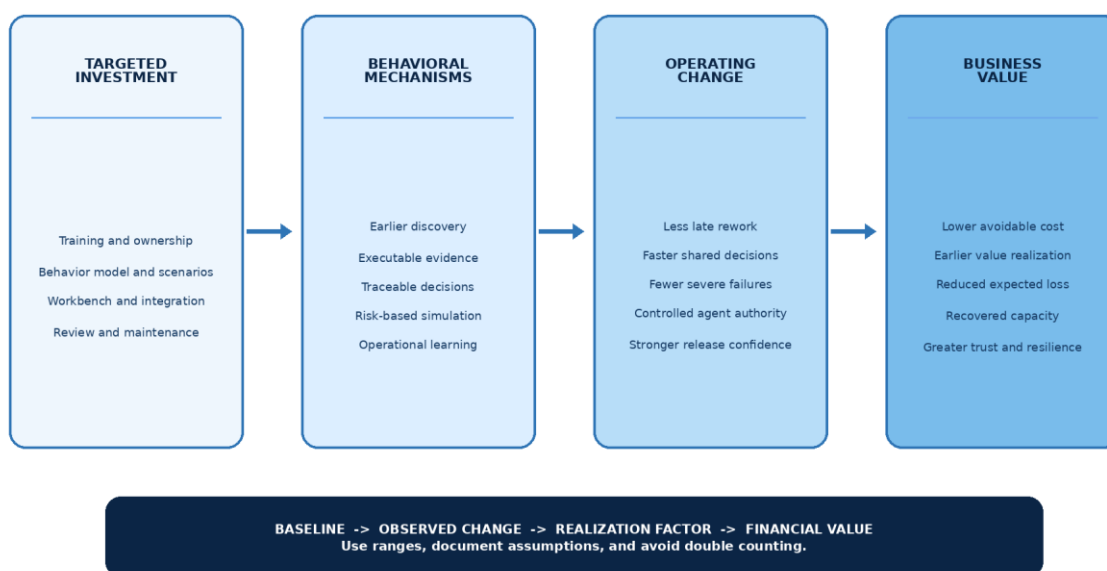


Figure 1. A defensible SDSE business case connects investment to mechanism, observable change, and financial value.

The value is decision-specific

A model that does not change a decision, reduce uncertainty, improve evidence, or protect an important outcome may have little economic value. The SDSE case should therefore start with a consequential decision: whether an agent receives authority, whether a workflow is safe under partial failure, whether an interface contract supports recovery, whether a release meets its behavioral promise, or whether a proposed control creates unacceptable side effects.

Proportionality rule

Do not model everything. Invest in behavioral fidelity and scenario coverage in proportion to consequence, uncertainty, interaction complexity, autonomy, frequency of change, and the cost of discovering an error later.

4. Five Sources of SDSE Business Value

4.1 Avoided rework

Behavioral ambiguity propagates. A misunderstood requirement becomes architecture, code, tests, prompts, controls, documentation, and operational procedures. SDSE can expose the misunderstanding before it multiplies. The measurable benefit is the portion of late rework that is plausibly prevented by earlier behavioral discovery.

4.2 Faster decisions and delivery

Executable scenarios give product, domain, engineering, security, reliability, and operations stakeholders a shared object to challenge. This can shorten debates driven by conflicting interpretations and reduce late approval delay. The benefit is not simply faster coding; it is faster convergence on a decision that deserves implementation.

4.3 Reduced operational loss

Simulation can reveal unsafe sequences, boundary conditions, failure interactions, and recovery defects before they occur in production. The financial benefit is risk-adjusted: the expected annual loss before SDSE minus the expected annual loss after the validated controls are implemented.

4.4 Controlled AI and agent autonomy

AI systems can generate changes and take actions faster than humans can review each artifact. SDSE provides a behavioral boundary against which authority, escalation, approval, override, and stop conditions can be evaluated. NIST's AI Risk Management Framework emphasizes incorporating trustworthiness into design, development, use, and evaluation, while its Playbook calls for metrics that show whether systems are fit for purpose and functioning as claimed [5][6].

4.5 Trust, governance, and business confidence

Leaders approve consequential releases more confidently when the promise, scenarios, evidence, model limits, residual risk, and accountable decision are visible together. This value may appear in reduced approval time, stronger audit readiness, fewer unresolved assumptions, and less executive escalation.

Value source	What to measure	Likely evidence
Avoided rework	Hours of late correction prevented	Tickets, change records, time data, retrospectives
Decision and delivery speed	Time removed from consequential decisions or release delay	Approval history, lead-time data, decision logs
Operational risk	Reduction in expected annual loss	Incidents, service credits, outage and support cost
AI control	Reduction in unsafe or unauthorized actions	Agent traces, policy events, overrides, stop events
Trust and governance	Lower review effort and stronger evidence coverage	Audit effort, exceptions, approval and trace records

Table 2. Each value claim should have a measurable operational expression and an evidence source.

5. The SDSE ROI Model

The model below is intentionally simple enough for an engineering leader to build with finance in a spreadsheet. It separates gross benefit from investment and distinguishes estimated capacity from cash savings.

5.1 Benefit calculations

Avoided rework value

Avoided Hours x Fully Loaded Hourly Cost x Realization Factor

The realization factor recognizes that time saved becomes value only when it is redeployed, reduces purchased labor, avoids hiring, or accelerates useful work.

Expected annual loss

Expected Events per Year x Average Business Impact per Event

For low-frequency events, use a probability range and document the evidence behind it.

Risk-reduction value

Baseline Expected Annual Loss - Residual Expected Annual Loss

Do not treat every simulated failure as a production incident that would certainly have occurred.

Delivery acceleration value

Monthly Incremental Contribution or Avoided Cost x Months Earlier x Confidence Factor

Use contribution margin or verified avoided cost rather than gross revenue when possible.

Recovered operating capacity

Hours Recovered x Fully Loaded Hourly Cost x Realization Factor

Report capacity separately from cashable savings unless the financial consequence is explicit.

5.2 Investment calculations

Cost category	Include
Capability creation	Training, coaching, method definition, initial standards, and enablement.
Pilot modeling	Workshops, Candidate Specification, BSN model, scenarios, EXPECT outcomes, simulation, and review.
Tooling and integration	BSN Workbench environment, repositories, automation, traceability, pipeline, and telemetry integration.
Ownership and governance	Behavior Model Owner allocation, domain participation, evidence review, approval, and risk acceptance.
Ongoing maintenance	Model changes, regression simulation, operational learning, periodic review, and support.
Transition and opportunity cost	Time diverted from other work, temporary delivery friction, and adoption support.

Table 3. A full-cost business case includes people, integration, governance, and maintenance.

5.3 Decision metrics

Net benefit

$$\text{Total Quantified Benefits} - \text{Total SDSE Investment}$$

Return on investment

$$\text{ROI} = (\text{Total Benefits} - \text{Total Investment}) / \text{Total Investment} \times 100\%$$

Simple payback period

$$\text{Payback Months} = \text{Total Investment} / \text{Annual Benefits} \times 12$$

For recurring costs, finance may prefer monthly net cash flow or a discounted cash-flow model.

Optional multi-year value

$$\text{NPV} = \text{Sum of Discounted Net Cash Flows} - \text{Initial Investment}$$

Use the organization's approved discount rate and separate one-time from recurring costs.

5.4 Classify the benefit before adding it

Benefit class	Financial treatment	Example
Cashable	Reduces actual spend or creates measurable contribution.	Avoided contractor cost, service credits, penalties, or infrastructure expense.
Capacity	Returns time that can be redeployed but does not automatically reduce budget.	Engineering hours moved from rework to planned delivery.
Risk-adjusted	Reduces the expected value of uncertain loss.	Lower probability or impact of a severe behavioral incident.

Benefit class	Financial treatment	Example
Strategic	Improves confidence, option value, learning, or readiness.	Ability to approve an AI capability safely or respond faster to change.

Table 4. Separating benefit classes prevents capacity and risk estimates from masquerading as guaranteed cash savings.

6. Risk-Adjusted Value and Sensitivity

A point estimate can create false confidence. GAO guidance recommends sensitivity analysis so decision makers can see a range of outcomes and understand which assumptions drive the estimate [3]. An SDSE business case should therefore include at least a conservative, expected, and higher-value scenario.

Risk-adjusted reasoning

For frequent events, historical event counts and average impact may provide a useful baseline. For rare but severe outcomes, estimate a probability range, impact range, and confidence level. Record whether SDSE is expected to reduce likelihood, reduce impact, or both. The estimate should reflect residual risk after the modeled controls are implemented - not assume that simulation eliminates failure.

The most important question

Which assumption changes the decision? If a modest change in incident reduction, realization factor, hourly cost, or adoption effort turns the case negative, leaders need to see that sensitivity before approving scale.

Recommended scenario structure

Scenario	How to construct it
Conservative	Use lower benefit realization, smaller risk reduction, slower adoption, and higher implementation effort.
Expected	Use the most supportable values from internal history, pilot evidence, and accountable expert judgment.
Higher value	Use plausible favorable outcomes, not theoretical maximums; retain full costs and residual risk.

Table 5. Scenarios should vary the assumptions that actually drive the result.

Sensitivity is not pessimism. It is a test of decision quality. A pilot can be worthwhile even when the conservative financial case is negative if it buys important evidence about a strategic, regulated, safety-relevant, or high-autonomy capability. The decision maker should know that this is an evidence investment rather than disguise it as guaranteed savings.

7. Illustrative Business Case: Autonomous Production Remediation

The following example demonstrates the method. It is illustrative and does not represent a customer result or a guaranteed outcome. An organization is considering an AI agent that can diagnose incidents, restart workloads, scale capacity, reroute traffic, and initiate rollback. Leaders want faster recovery, but the agent's authority creates the possibility of cascading remediation.

7.1 Baseline and proposed SDSE intervention

The team selects the agent's cross-region remediation decision as the pilot behavior. It creates a Candidate Specification, models agent authority and operating state in BSN, simulates delayed telemetry, partial dependency failure, capacity saturation, repeated restart, rollback failure, and operator intervention, then approves a behavioral baseline with cooldown, blast-radius, evidence, approval, and stop conditions.

Baseline input	Illustrative value	Evidence source
Severe behavior-related incidents	4 per year	Incident and post-incident records
Average direct impact	\$180,000 per severe event	Service credits, lost contribution, support, recovery
Behavior-driven rework	1,800 hours per year at \$115/hour	Tickets, time sampling, release retrospectives
Incident engineering effort	600 hours per year at \$125/hour	Incident staffing and time records
Governance and approval effort	300 hours per year at \$120/hour	Review and approval records
Delayed value opportunity	\$50,000 per month	Finance-approved contribution or avoided cost

Table 6. The baseline is explicit so reviewers can replace every illustrative input with internal data.

7.2 Expected first-year benefits

Benefit	Value	Calculation
Reduced expected incident loss	\$180,000	25% reduction x 4 events x \$180,000
Avoided rework capacity	\$43,470	1,800 h x \$115 x 30% reduction x 70% realization
Recovered incident capacity	\$10,500	600 h x \$125 x 20% reduction x 70% realization
Earlier value realization	\$56,250	\$50,000/month x 1.5 months x 75% confidence
Reduced review effort	\$6,300	300 h x \$120 x 25% reduction x 70% realization
Total quantified benefit	\$296,520	Sum of quantified first-year benefits

Table 7. Capacity benefits use a realization factor; incident prevention is risk-adjusted rather than treated as certain.

7.3 Expected first-year investment

Investment element	Illustrative cost
Training and coaching	\$24,000
Cross-functional modeling and simulation	\$36,000
Workbench, repository, pipeline, and telemetry integration	\$30,000
Behavior Model Owner allocation	\$60,000
Ongoing model maintenance and evidence review	\$36,000
Total first-year investment	\$186,000

Table 8. The example includes the labor and operating costs frequently omitted from tooling-led cases.

7.4 Result and sensitivity

Expected first-year result

Net Benefit = \$110,520 | ROI = 59.4% | Simple Payback = 7.5 Months

Scenario	Benefits	Investment	ROI	Decision implication
Conservative	\$160,000	\$195,000	-17.9%	Collect more evidence or narrow the pilot.
Expected	\$296,520	\$186,000	59.4%	Proceed if non-financial risks are acceptable.
Higher value	\$510,000	\$205,000	148.8%	Validate drivers before using for funding.

Table 9. Illustrative sensitivity exposes the decision to benefit realization and adoption cost.

Case lesson

The expected case is attractive, but the conservative case is negative. A responsible decision is to fund a bounded pilot with predefined evidence gates, not to claim that the ROI is already proven.

8. Building a Business Case Leaders Will Accept

The strongest SDSE case is not the one with the largest percentage. It is the one whose assumptions are visible, whose evidence can be checked, and whose decision can be revised when new information arrives.

1. **Select a consequential behavior.** Choose a decision, interaction, control, recovery path, or autonomous action with meaningful uncertainty and measurable business consequence.
2. **Define the alternatives.** Compare at least the current approach, a targeted SDSE pilot, and any credible non-SDSE control improvement. Do not compare SDSE only with doing nothing.
3. **Establish the baseline.** Use incident history, rework records, release delays, approval time, support cost, customer impact, and finance-approved value measures.
4. **Describe the SDSE mechanism.** State what will be modeled, which scenarios will be executed, what evidence will be produced, and which decision that evidence will improve.
5. **Estimate the full investment.** Include training, participation, tooling, integration, ownership, review, maintenance, and opportunity cost.
6. **Quantify benefits conservatively.** Apply realization, probability, confidence, and attribution factors; classify cash, capacity, risk, and strategic value separately.
7. **Test sensitivity.** Vary the assumptions most likely to change the result and show conservative, expected, and higher-value cases.
8. **Define evidence gates.** Agree in advance what findings, measures, and thresholds will justify scaling, revision, more observation, or stopping.
9. **Assign owners and approval.** Name the owner of every material assumption and the people authorized to accept the model, estimate, residual risk, and funding decision.
10. **Update after the pilot.** Replace assumptions with observed evidence and reissue the business case before enterprise scale.

Finance-ready evidence record

Field	Required content
Metric	Exact definition, unit, inclusion rules, and exclusions.
Baseline	Period, system scope, source, data quality, and owner.

Field	Required content
Change mechanism	Why the SDSE activity is expected to change the metric.
Attribution	What portion of observed change can reasonably be assigned to SDSE.
Realization	What portion becomes cash, capacity, risk reduction, or strategic value.
Range	Conservative, expected, and higher-value inputs.
Review	Named engineering, finance, risk, and business reviewers.

Table 10. Every material estimate should be traceable to an evidence record.

9. Pilot Design and Measurement

The first SDSE implementation should be large enough to matter and small enough to measure. A pilot that targets a trivial behavior may succeed technically but fail to establish business value. A pilot that attempts to model an entire platform will take too long and blur attribution.

THE SDSE PILOT MEASUREMENT LOOP

Prove value on a bounded decision before asking the enterprise to scale.

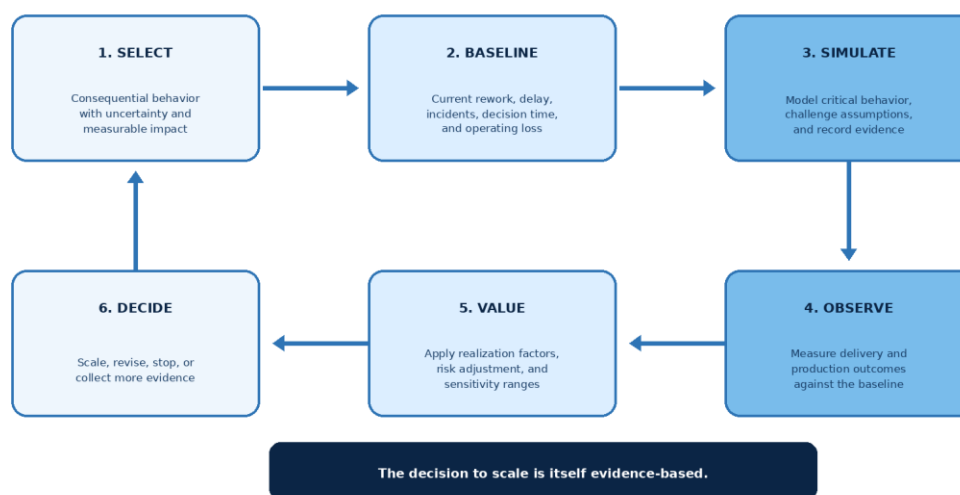


Figure 2. The pilot is a controlled learning investment whose final output is a better scaling decision.

Pilot selection scorecard

Score each criterion from 0 to 2: 0 means low or absent, 1 means moderate, and 2 means high. The score is a decision aid, not a statistical model. Favor pilots with consequence, uncertainty, available evidence, and an accountable sponsor.

Criterion	What a high score means
Business consequence	Failure or delay materially affects customers, revenue, cost, compliance, safety, or trust.
Behavioral ambiguity	Stakeholders hold different interpretations or important assumptions remain implicit.
Interaction complexity	Several services, agents, states, policies, or human workflows interact.
Autonomy	Software or AI can take consequential action without immediate human review.
Change frequency	The behavior or its environment changes often enough for reusable simulation to matter.
Baseline evidence	Historical data can support rework, delay, incident, effort, or value estimates.
Sponsor and owner	A business sponsor and Behavior Model Owner can act on the evidence.

Table 11. A strong pilot combines material consequence with measurable evidence and accountable ownership.

Practical threshold

A score of 10 or more out of 14 suggests a promising pilot candidate, provided the sponsor can define a real decision and finance can verify at least one important baseline measure.

10. Measures That Matter

The measurement system should balance delivery, quality, risk, and economic realization. DORA's software delivery measures remain useful for overall flow and stability, but an SDSE pilot also needs behavior-specific measures. The objective is to observe whether SDSE improved a consequential decision or outcome, not whether the team produced more artifacts [1].

Measure	Observable evidence
Earlier discovery	Material ambiguities, unsafe interactions, weak assumptions, and missing scenarios found before code or deployment.
Behavioral decision latency	Elapsed time from a consequential behavioral question to evidence-backed acceptance, revision, or rejection.
Avoided rework	Late requirement, design, integration, security, release, and incident work plausibly prevented by earlier simulation.
Change failure and recovery	Behavior-related change failures, customer-impact minutes, and recovery effort.
Promise traceability	Critical behaviors linked to scenarios, expectations, architecture, code, tests, controls, deployment, and telemetry.

Measure	Observable evidence
Human-control coverage	Consequential agent actions with modeled authority, approval, escalation, override, and stop conditions.
Model-reality learning	Time from a material operational observation to model revision, resimulation, and an approved new baseline.
Financial realization	Verified cash, redeployed capacity, avoided expected loss, and earlier value attributable to the intervention.

Table 12. Measures should connect behavioral assurance to delivery and business outcomes.

Measurement rules

11. Define the measure before the pilot and freeze the baseline period unless a documented reason requires change.
12. Use the same system scope, severity rules, and cost treatment before and after the intervention.
13. Record other changes that could explain the result, including staffing, architecture, traffic, process, and tooling changes.
14. Use attribution and realization factors when SDSE is one contributor among several.
15. Report both favorable and unfavorable evidence, including new costs or delays introduced by the method.

11. What Different Leaders Need to Hear

Audience	Lead with	Question to answer
CEO or business sponsor	Customer outcomes, earlier value, strategic risk, and confidence in consequential change.	Which business promise becomes safer or faster?
CFO	Full costs, benefit classes, assumptions, ranges, payback, and sensitivity.	Which inputs are verified and which remain judgment?
CIO or CTO	Delivery confidence, architecture decisions, AI governance, reuse, and reduced rework.	How does the living model improve the engineering system?
Software leader	Earlier clarification, fewer late changes, stronger tests, and less incident interruption.	Which workflow and team measures will change?
Risk, security, or compliance	Traceability, authority, evidence, residual risk, and controlled approval.	What risk is reduced, what remains, and who accepts it?
Product leader	Faster convergence on intended outcomes and fewer acceptance surprises.	How will customers experience the improvement?

Table 13. The evidence stays consistent; the decision lens changes by audience.

Do not present SDSE as a request to fund more modeling. Present it as a controlled investment in better decisions about software behavior. Business leaders do not need to understand every notation construct. They do need to understand the promise at risk, the decision being improved, the cost of being wrong, the evidence the pilot will produce, and the conditions for scaling.

12. Common Mistakes and Objections

Mistakes that destroy credibility

- 16.Counting every simulated defect as a production incident that was certainly prevented.
- 17.Applying a generic productivity percentage to all engineering labor.
- 18.Counting recovered hours as cash savings without a realization plan.
- 19.Counting the same avoided work under productivity, rework, acceleration, and incident reduction.
- 20.Omitting model ownership, integration, review, and maintenance costs.
- 21.Using industry averages when current internal evidence exists.
- 22.Presenting one precise ROI number without ranges or sensitivity.
- 23.Scaling before the pilot replaces important assumptions with observed evidence.

Common objections

Objection	Business response
"Simulation will slow delivery."	Targeted simulation adds early effort but can reduce expensive late correction. The pilot should measure the net effect on decision and delivery time.
"We already test extensively."	Testing proves implementation against configured expectations. SDSE challenges whether the behavioral promise and those expectations are right before implementation.
"Avoided incidents cannot be proven."	They cannot be treated as certain. Use expected-loss ranges, leading evidence, residual risk, and sensitivity.
"The model will become stale."	It will if treated as documentation. A named owner, material-change triggers, operational evidence, and versioned baselines make it a living asset.
"We cannot model everything."	SDSE does not require completeness. Fidelity and scenario coverage follow risk, uncertainty, and consequence.
"Open-source tooling means the program should be free."	License cost is only one input. Training, domain participation, integration, governance, ownership, and maintenance remain real investments.

Table 14. The response to skepticism is measurable evidence, not advocacy alone.

13. Decision Framework

At the end of the pilot, leaders should make one of four explicit decisions. The choice should be based on the evidence and the consequence of uncertainty, not on whether the team enjoyed the modeling exercise.

Decision	When it is justified
Scale	Observed value is material, the mechanism is credible, operating ownership exists, and sensitivity remains acceptable.
Revise	The behavior matters, but the model, workflow, integration, or cost structure must change before expansion.
Collect more evidence	The pilot improved understanding, but baseline quality, attribution, or observation time is insufficient.
Stop	The benefit is immaterial, the mechanism is weak, the cost is disproportionate, or another control is superior.

Table 15. Stopping a weak case is a successful governance outcome, not a failed pilot.

Minimum approval package

24. Defined business promise and pilot decision.
25. Candidate Specification, model scope, scenarios, evidence, and fidelity statement.
26. Baseline and observed measures with sources and owners.
27. Full cost, quantified benefits, benefit classes, and realization plan.
28. Conservative, expected, and higher-value scenarios with key drivers.
29. Residual risks, exclusions, alternatives, and accountable acceptance.
30. Recommended scale, revise, observe, or stop decision.

Approval principle

Fund the next increment of evidence and value. Do not authorize enterprise scale merely because the method is compelling in theory.

Conclusion

The economic value of Simulation-Driven Software Engineering does not come from producing more diagrams. It comes from improving consequential decisions before misunderstanding becomes embedded in architecture, code, tests, controls, deployments, operating procedures, and customer experience.

A credible SDSE business case begins with a bounded behavior and a measurable baseline. It identifies the mechanism by which simulation will change an outcome, includes the full investment, distinguishes cash from capacity and risk-adjusted value, applies realization and confidence factors, tests sensitivity, and defines evidence gates before the pilot begins. It then replaces assumptions with observed results and asks leaders to scale only what has demonstrated value.

This approach respects both engineering and financial reality. Software behavior is uncertain. Prevented losses cannot be claimed with certainty. Recovered time is not automatically cash. Models require ownership and maintenance. Yet the cost of late behavioral discovery is real, and AI increases the speed at which weak assumptions can become functioning systems. SDSE gives organizations a disciplined way to buy understanding while change is still affordable.

Final proposition

The purpose of the SDSE business case is not to prove that simulation is always valuable. It is to show, with evidence, where better behavioral decisions are worth more than the cost of producing them.

Appendix A. SDSE Business Case Worksheet

Use this worksheet to prepare a pilot proposal. Replace every placeholder with an owned statement or value. If the data is unknown, label it as an assumption and define how the pilot will resolve it.

Business-case field	Working entry
Business promise	[What important outcome must the software produce or protect?]
Consequential decision	[What funding, authority, design, release, or control decision will SDSE improve?]
System and behavior scope	[Which participants, states, interactions, conditions, and outcomes are included?]
Alternatives	[Current approach; targeted SDSE; other viable control or engineering alternative.]
Baseline period and evidence	[Dates, systems, records, measures, data quality, and owners.]
Candidate Specification	[Intent, rules, constraints, unacceptable outcomes, authority, and assumptions.]
Simulation plan	[Normal, boundary, failure, timing, misuse, recovery, and human-control scenarios.]
Expected mechanism	[How will the intervention reduce rework, delay, expected loss, or decision uncertainty?]
Investment	[Training, modeling, integration, ownership, review, maintenance, and opportunity cost.]
Benefits	[Cashable, capacity, risk-adjusted, and strategic benefits - shown separately.]
Realization and confidence	[Attribution, realization factor, probability, and confidence factor.]
Sensitivity	[Conservative, expected, and higher-value values for the principal drivers.]
Evidence gates	[Thresholds for scale, revise, collect more evidence, or stop.]
Accountability	[Business sponsor, Behavior Model Owner, finance reviewer, risk owner, approval authority.]

Table 16. A reusable worksheet for preparing a decision-ready SDSE pilot proposal.

References

- [1] Google Cloud DORA, "Announcing the 2025 DORA Report," September 23, 2025. [2025 DORA report announcement](#)
- [2] NASA, "Cost Effectiveness Considerations," NASA Systems Engineering Handbook reference, updated July 26, 2023. [NASA lifecycle cost guidance](#)
- [3] U.S. Government Accountability Office, Cost Estimating and Assessment Guide: Best Practices for Developing and Managing Program Costs, GAO-20-195G, March 2020. [GAO Cost Estimating Guide](#)
- [4] U.S. Government Accountability Office, Information Technology Cost Estimation: Agencies Need to Address Significant Weaknesses in Policies and Practices, GAO-12-629, July 2012. [GAO IT cost-estimation report](#)
- [5] National Institute of Standards and Technology, "Artificial Intelligence Risk Management Framework," accessed August 15, 2026. [NIST AI RMF](#)
- [6] National Institute of Standards and Technology, "AI RMF Playbook - Measure," accessed August 15, 2026. [NIST measurement guidance](#)
- [7] M. Souppaya, K. Scarfone, and D. Dodson, Secure Software Development Framework (SSDF) Version 1.1, NIST SP 800-218, February 2022. [NIST SP 800-218](#)
- [8] Marc Hornbeek, Debashis Bhattacharyya, and Jitin Dhanani, Simulation-Driven Software Engineering: A Behavior-First Discipline for Modern Systems, 2026. [Purchase the SDSE book on Amazon](#)

About the Author

Marc Hornbeek, a.k.a. DevOps_the_Gray, is an award-winning engineer, author, educator, executive consultant, and keynote speaker with five decades of experience across systems engineering, packet networks, software development, behavioral simulation, testing, automation, DevOps, DevSecOps, SRE, platform engineering, and intelligent software delivery. His early engineering work included integrated tools for packet-network simulation, performance, conformance, observation, and test automation. He is the originator of Simulation-Driven Software Engineering and leads the development of Behavior Simulation Notation and the BSN Workbench. Marc received the IEEE Region 6 Outstanding Engineer Award in 2016 and has authored multiple books and professional certification courses. He founded PrimaModus® to help organizations model, simulate, validate, and engineer software that behaves as intended.

Contact: mhornbeek@engineeringdevops.com