WHAT IS THE CHARTER BUDGET?

The **Charter Budget** feature in MyScrumBudget™ adds a statistically defensible contingency to the project ETC, producing a risk-adjusted budget for the project charter. ETC reflects planned team allocations and is treated as the **expected (mean) cost** of delivering the chartered scope; actual costs will vary around it. The model derives a **coefficient of variation (CV)** from five observable project characteristics and calculates a charter budget at a chosen probability percentile. Project-type base CVs are **reference-class calibrations** — uncertainty levels consistent with the cost variability documented for similar completed projects in the research literature (see Research Basis, page 2). An optional **optimism-bias adjustment** (off by default) lets teams whose allocations skew optimistic raise the cost basis by a stated percentage before contingency is applied — shown as a separate, traceable line (see page 4). Three probability distributions are available (Normal, Lognormal, Beta-PERT) and are described on page 3. When ETC comes from a P80 schedule forecast (e.g., SPERT® Forecaster), see page 5 for the correct cost target percentile. *Example: a $164,834 ETC (custom software, neutral risk profile, CV = 25%, no optimism adjustment) yields σ = $41,209. Under Normal: P80 = $199,535 (+21.1%). Under Lognormal: P80 = $196,695 (+19.3%).*

$CV_{composite}$ = Base CV (project type) + Δrequirements + Δexperience + Δorg change + Δintegration
$CV_{applied}$ = clamp($CV_{composite}$, floor 8%, ceiling 50%)
Cost basis = ETC × (1 + optimism uplift) [uplift = 0 by default → basis = ETC] σ = Cost basis × $CV_{applied}$
Charter Budget (P_n) = f(Cost basis, $CV_{applied}$, p, distribution) — Normal: Cost basis + z_p × σ (see page 3 for Lognormal and Beta-PERT)
P70: z = 0.524 P80: z = 0.842 P90: z = 1.282 P95: z = 1.645 [z-scores for Normal; Lognormal and Beta-PERT compute differently]

■ **How to access:** On the project detail page, click **Set charter budget** on the Baseline Budget card. The ETC from the active reforecast pre-populates as the most likely cost — no re-entry needed. Complete the five-question risk profile, choose a distribution and target percentile, review the live probability chart, and click **Apply as baseline budget**. The Baseline Budget card also supports direct inline editing for externally assigned budgets — the risk model is optional, not required.

PROJECT TYPE — BASE COEFFICIENT OF VARIATION

Project type is the primary uncertainty driver. The base CV is a reference-class calibration reflecting documented cost variability for similar completed projects. Select the type that best describes the *primary* nature of the work. P80 contingency values below assume Normal distribution and neutral adjustment factors.

Project type	Base CV	P80 contingency above ETC
Vendor / COTS implementation	15%	+12.6%
Infrastructure / cloud migration	18%	+15.2%
Business process / org transformation	22%	+18.5%
Custom software development	25%	+21.1%
Data engineering / analytics / BI	25%	+21.1%
AI / ML / data science	40%	+33.7%*

*P80 contingency = 0.842 × Base CV (plain-normal reference, neutral adjustments). Lognormal yields modestly lower P80; Beta-PERT yields modestly higher. Adjustment factors (page 2) will raise or lower the final charter budget from the values shown. *****The app computes a Normal distribution **truncated at $0**; truncation is negligible at low CV (Custom 25% still rounds to +21.1%) but lifts the AI/ML 40% figure to +33.8% in-app — the +33.7% shown is the plain-normal reference.*

ADJUSTMENT FACTORS

Four dimensions adjust the base CV; all adjustments are **additive**. Additive summation is a deliberate **positively-correlated (ρ = 1) assumption** — IT risk drivers tend to co-occur, so we add them rather than combining them in quadrature (which would assume independence and produce a less conservative result). $CV_{composite}$ is clamped to a **floor of 8%** (irreducible execution uncertainty — even the most favorable profile carries residual risk) and a **ceiling of 50%** — a governance signal that project scope may not yet be stable enough to charter, not merely a statistical threshold. **When the ceiling is active, the app flags it to the user:** a clamped CV is the least-trustworthy output and signals that the project may warrant re-scoping before chartering rather than a larger contingency. Requirements asymmetry (+8% exploratory vs. –5% well-defined) is supported by behavioral economics (Prospect Theory) and scope creep research: vague requirements create proportionally more upside risk than clear requirements reduce downside. *Org change and integration CV adjustments are expert-calibrated priors — see Research Basis for detail.*

Factor	Favorable	Neutral	Unfavorable
Requirements clarity	Well-defined CV: –5%	Partially defined CV: ±0%	Exploratory CV: +8%
Team experience	Experienced CV: –5%	Some experience CV: ±0%	New territory CV: +7%
Org change impact	Low CV: –3%	Moderate CV: ±0%	High CV: +6%
Integration complexity	Standalone CV: –3%	Moderate CV: ±0%	High complexity CV: +6%

RESEARCH BASIS

McConnell (2006)	**Software Estimation: Demystifying the Black Art.** The Cone of Uncertainty places estimates at **initial concept** in a **0.25×–4× band** (roughly –75% / +300%, an asymmetric ~16× span) — a project charter sits at this *wide* end. (The narrower ±50% band applies to a later milestone, after product definition is approved.) Custom software and data engineering base CVs (25%) are reference-class calibrations broadly consistent with early-phase uncertainty of this order. Note: the Cone describes estimate-range compression over project phases, **not** a probability distribution, and is not used here as one.
Flyvbjerg & Budzier (2011, 2022)	**Why Your IT Project May Be Riskier Than You Think** (HBR, 2011). One in six IT projects exceeds a 200% cost overrun; average overrun 27%. Flyvbjerg et al. (2022, n = 5,392) confirm a **power-law distribution** — many moderate overruns plus a fat right tail (headline tail exponent $\alpha \approx 2.3$; finite mean, with infinite variance only deep in the tail). The widely-cited "18% exceed 50%, averaging 447%" figure is from *How Big Things Get Done* (2023), not the 2022 paper. Anchors AI/ML (40%) and business process (22%) base CVs. Flyvbjerg's **Reference Class Forecasting (RCF)** — grounding budgets in the actual cost distribution of similar completed projects — provides the conceptual framework for the base CV table.
Boehm — COCOMO II	Experience cost drivers (APEX 1.22–0.81, PEXP 1.19–0.85, LTEX 1.20–0.84) imply roughly ±20% effort shifts across the experience range. Team experience CV adjustments (–5% experienced, +7% new territory) are conservative subsets applied to *uncertainty* rather than to the *point-estimate* effort multipliers COCOMO II actually defines — the analogy is directional, not a derivation.
PMI Pulse of the Profession	Annual survey: 40–45% of projects miss original budget goals; poor requirements definition is a leading cause. Supports the requirements clarity factor. Asymmetry (+8% exploratory vs. –5% well-defined) aligns with scope creep research and Prospect Theory findings in behavioral economics.

Prosci / McKinsey	Change-management effectiveness is a top-three predictor of IT project budget performance, particularly for business-process and ERP work. *These studies measure success **likelihood** and adoption outcomes — not cost-variance distributions — so they provide only directional support. The org-change CV magnitudes (–3% / +6%) are expert-calibrated priors, not figures these sources quantify.*
Jørgensen (multiple)	Research on software effort estimation in Agile environments finds that iterative delivery improves visibility and adaptation but does not inherently reduce project-level cost variance. Requirements uncertainty and integration complexity remain active risk factors regardless of development method — supporting unchanged base CVs for Scrum projects rather than applying an Agile discount.

Distribution and tail-risk note: Real IT project costs follow a power-law pattern — many moderate overruns plus a rare but extreme right tail (Flyvbjerg, 2022). Normal and Beta-PERT both understate this tail at CV > 30%; Lognormal better captures right-skew behavior and is preferred for projects with heavy overrun exposure (see page 3). For AI/ML or highly exploratory work, **P85** is recommended regardless of distribution. **Stakeholder defense:** contingency is derived from research-calibrated project characteristics — each of the five risk profile selections is the explicit, traceable analytical basis for the charter budget amount.

CHOOSING AN UNCERTAINTY MODEL

Three probability distributions are available. All three use the same $CV_{applied}$ and $\sigma = ETC \times CV_{applied}$ — they differ only in how that uncertainty is translated into a charter budget at the chosen percentile. **Normal** (truncated at $0, default) is appropriate when ETC is an unbiased estimate and overruns and underruns are equally plausible. **Lognormal** is preferred when most outcomes are expected below ETC — evaluation projects, research sprints, or proof-of-concept work where team members rarely consume their full allocation, but occasional large overruns remain possible. **Beta-PERT** suits bounded-scope projects where a hard upper cost limit can be reasonably defined (fixed-price contracts, regulated cost caps). The distribution choice does not affect CV computation or the questionnaire.

Distribution	When to choose	P50 vs. ETC	P80 at CV = 25%, neutral
Normal (default)	ETC is your unbiased best estimate; over- and underruns equally plausible; general-purpose default for most projects	P50 ≈ ETC	+21.1% ($199,535 on $164,834 ETC)
Lognormal	Most outcomes expected below ETC; rare but large overruns possible — evaluation, research, PoC, or any project where full allocation is rarely consumed	P50 < ETC (median below most likely cost)	+19.3% at P80; higher than Normal at P95 and above
Beta-PERT	Bounded scope with a hard upper cost limit (fixed-price vendor, regulatory cap, or ring-fenced budget)	P50 = ETC (symmetric by construction)	+23.0% (bounded distribution concentrates mass toward center)

Normal (Truncated, default): $\sigma = ETC \times CV$ $Q(p) = ETC + z_p \times \sigma$ [bounded at $0; z_p values on page 1]

Lognormal: $\sigma_{ln} = \sqrt{\ln(1 + CV^2)}$ $\mu_{ln} = \ln(ETC) - \sigma_{ln}^2 \div 2$ $Q(p) = \exp(\mu_{ln} + z_p \times \sigma_{ln})$

Median = $\exp(\mu_{ln})$ < ETC — more than half of outcomes fall below the most likely cost

Beta-PERT ($\alpha_1 = \alpha_2 = 3$, symmetric): $O = \max(0, ETC - \sqrt{7} \times \sigma)$ $P_b = ETC + \sqrt{7} \times \sigma$ $Q(p) = O + (P_b - O) \times Beta_{3,3}(p)$

Beta(3,3) CDF: $F(x) = 10x^3 - 15x^4 + 6x^5$ [solved by bisection] If CV > 38%: O clips to $0 and $P_b = 2 \times ETC$

■ **Lognormal for under-budget-likely projects:** A lognormal distribution with mean = ETC places its *median below ETC* — more than half of all modeled outcomes fall under the planned cost. This accurately reflects evaluation projects, research sprints, or initiatives where full team allocation is rarely consumed. The lower P80 charter budget (e.g., +19.3% vs. +21.1% at CV = 25%) is not a sign of less protection — the lognormal right tail provides stronger coverage at P90 and above, where actual worst-case overruns tend to occur. The app displays the P50 (median) alongside the charter budget to make this behavior visible.

CV > 30% — consider P85: At higher CV values, all three distributions understate extreme tail risk relative to the empirical power-law behavior of IT project costs. When $CV_{applied}$ > 30%, the app suggests selecting **P85** as the target percentile to partially compensate. For AI/ML or highly exploratory projects, P85 is the recommended default regardless of CV. **Beta-PERT edge case:** when CV exceeds approximately 38%, the lower bound clips to $0 and the distribution becomes right-skewed; the upper bound shifts to 2×ETC to preserve mean = ETC.

MODEL CHOICES & LIMITATIONS

This model is a transparent **parametric contingency heuristic**, not a guarantee. It translates research-calibrated project characteristics into an explainable percentile budget — every dollar traces to a risk-profile selection. The choices below are deliberate; understanding them is part of defending the number to a sponsor.

What ETC represents, and the optional optimism-bias adjustment

ETC is treated as the **expected (mean) cost** of the chartered scope under current allocations. The contingency layer ($z \times \sigma$) adds *spread* around that center — the five questionnaire factors are calibrated to burn-rate and scope variance for a fixed plan, not to schedule risk. The estimation literature (Flyvbjerg; Jørgensen) warns that plan-based point estimates are often themselves biased low. The model handles this with two explicit options rather than by silently inflating CV:

Option 1 Default (no adjustment)	ETC is taken as an unbiased expected cost. Optimism uplift = 0; the cost basis equals ETC exactly. Appropriate for disciplined Scrum delivery where allocations reflect realistic burn rates. **This is the default and requires no input.**
Option 2 Opt-in uplift (%)	For teams whose allocations skew optimistic, an optional percentage raises the cost basis *before* contingency: **Cost basis = ETC × (1 + uplift)**, and σ scales off the adjusted basis. The uplift is a **user-supplied judgment, not a research-derived constant**, and is shown as a separate, labeled line so the sponsor sees the center correction and the contingency distinctly.

■ **Two honest lines beat one inflated CV.** When an uplift is set, the results read as: Expected cost (ETC) → + *optimism adjustment* → Adjusted cost basis → + *risk contingency at* P_n → Charter Budget. With the default (uplift = 0) the two middle lines disappear and the math is identical to a pure spread-only model.

Deliberate assumptions and known limitations

Additive CV (ρ = 1)	Factor deltas are **added**, which assumes risk drivers are positively correlated — a conservative choice. Independent (root-sum-square) addition would roughly halve the worst-case band and is almost certainly too low, since IT risk drivers genuinely co-occur.
50% ceiling (flagged when active)	The ceiling is a **governance signal**, not catastrophe insurance. When it binds (e.g., AI/ML with multiple unfavorable factors), the app warns the user: a clamped CV is the least-trustworthy output and usually means the scope should be re-defined before chartering. P80 contingency cannot exceed +42% under the ceiling.
Thin-tailed distributions	Real IT costs follow a **power-law** with a fat right tail that *no* Normal, Lognormal, or Beta-PERT fully captures. Manage extreme tail risk through floor/ceiling/percentile choice and explicit caveats — not by inflating z. Cornish-Fisher skew correction is **deliberately not used**: at the P80 default ($z < 1$) its skew term is negative and would *lower* contingency.
Coverage boundaries	Project **size/duration** and **schedule compression** — first-order cost drivers in COCOMO — are not yet modeled; base CVs assume typical-sized projects. Calibrated for **disciplined Scrum delivery**, not waterfall or fixed-scope vendor megaprojects.

Fixed-scope / Scrum caveat: This P80 assumes the chartered scope is delivered. In a fixed-budget Scrum model, cost and time are often held while **scope** flexes — so an overrun may surface as *reduced delivered scope* rather than a dollar overrun. The charter budget bounds the dollars, not the scope.

Roadmap (not in the current model): *a Reference Class Forecasting track seeded from the SPERT® Suite's accumulating real actuals (the most defensible end-state); a size/duration factor and schedule-compression flag; a documented surcharge and lifted ceiling for projects with three or more unfavorable factors; and charter-vs-actual back-testing once enough projects accumulate.*

SCHEDULE UNCERTAINTY AND THE CHARTER BUDGET

When the ETC pre-populated into this panel was calculated to a **P80 (or higher) completion date** — for example, from SPERT® Forecaster — the Most Likely Cost field already contains a schedule buffer expressed in dollars. It represents the cost of working to a date that only 20% of projects will exceed. Adding a P80 cost target on top creates a **double-stack**: because schedule and cost uncertainty are correlated (complex projects tend to overrun both), effective coverage rises to roughly **P88–P90** while the charter signals P80 to stakeholders. Two integration approaches resolve this.

Option A Split risk	**ETC input:** P80 schedule ETC from your forecasting tool (e.g., SPERT® Forecaster). **Cost target:** P60. **Effective coverage:** ≈ P80–P83. The schedule buffer embedded in the ETC does the heavy lifting; the cost model captures burn rate and scope variance around that extended timeline. Use when two tools are explicitly sharing risk work — the forecasting tool owns schedule risk, and the Charter Budget model owns cost-given-schedule risk.
Option B Single-tool ownership (recommended)	**ETC input:** P50 (most likely) schedule ETC — labor cost to the *expected* completion date, not the padded P80 date. **Cost target:** P80 (or your standard). **Effective coverage:** true P80. SPERT® Forecaster owns schedule risk; MyScrumBudget Charter Budget owns cost risk. The five CV adjustment factors measure burn rate and scope variance for a fixed timeline — when fed the most likely ETC, they do exactly what they were designed for. Cleaner conceptual model: one tool, one risk dimension.

■ **One-sentence rule:** If your ETC comes from a P80 schedule forecast, target P60. If your ETC comes from the most likely schedule, target P80. Never combine both at P80 — that produces P88–P90 coverage while signaling P80 to stakeholders.

In-app schedule source indicator: The Charter Budget panel displays a checkbox below the Most Likely Cost field: *"ETC based on P80+ schedule forecast (e.g., SPERT® Forecaster)"*. When checked, the app shows an amber recommendation to switch to P60 and highlights that option in the percentile selector. The indicator is advisory — the final percentile choice is always yours. The selection is stored with the charter budget profile so the recommendation reappears correctly when the panel is reopened for editing.

REFERENCES

Sources underpinning the base CVs, adjustment factors, and distribution choices. Consolidated from a multi-agent external peer review (May 2026) with adversarial fact-checking; claims that did not survive verification were dropped or corrected.

• Flyvbjerg, Budzier, Lee, Keil, Lunn & Bester (2022). *The Empirical Reality of IT Project Cost Overruns: Discovering a Power-Law Distribution.* J. Mgmt. Information Systems 39(3). — the decisive study on the fat-tailed cost distribution (n = 5,392).

• Flyvbjerg & Budzier (2011). *Why Your IT Project May Be Riskier Than You Think.* Harvard Business Review. — 1-in-6 projects exceed 200% overrun; 27% mean.

• Flyvbjerg (2006). *From Nobel Prize to Project Management: Getting Risks Right.* — Reference Class Forecasting; endorsed by HM Treasury.

• Flyvbjerg, Budzier, Aaen, Keil & Zottoli (2026). *The Uniqueness of IT Cost Risk: A Cross-Group Comparison of 23 Project Types.* — IT as a distinct tail-risk class.

• Flyvbjerg & Gardner (2023). *How Big Things Get Done.* — source of the "18% exceed 50%, avg 447%" megaproject figure (not the 2022 paper).

• AACE International RP 18R-97. *Cost Estimate Classification System.* — estimate accuracy keyed to scope-definition maturity; asymmetric ranges.

• AACE International RP 119R-21. *Cost Estimate Accuracy Range and Contingency Determination.*

• AACE International RP 42R-08. *Risk Analysis and Contingency Determination Using Parametric Estimating.*

• HM Treasury, Supplementary Green Book Guidance: *Optimism Bias* (Mott MacDonald, 2002). — development/equipment uplift up to +200% capex.

• McConnell, S. (2006). *Software Estimation: Demystifying the Black Art.* — Cone of Uncertainty; initial concept 0.25×–4×.

• Boehm et al. *COCOMO II Model Definition Manual.* — multiplicative effort drivers (APEX 1.22–0.81, PEXP 1.19–0.85, LTEX 1.20–0.84; SCED; E = 1.0997).

• Little, T. (2006). *Schedule Estimation and Uncertainty Surrounding the Cone of Uncertainty.* IEEE Software 23(3). — lognormal fit; cone does not narrow.

• Eveleens & Verhoef (2010). *The Rise and Fall of the Chaos Report Figures.* IEEE Software 27(1). — why Standish CHAOS figures are not reproducible.

• Jørgensen (multiple). *Realism in Assessment of Effort Estimation Uncertainty*; *Eliminating Over-Confidence.* — nominal 90% intervals capture truth only 60–70% of the time.

• Jørgensen & Grimstad. *Over-Optimism in Software Development Projects: The Winner's Curse.*

• Moløkken-Østvold & Jørgensen (2003/2004). *A Review of Surveys on Software Effort Estimation.* — 30–40% mean overrun; flexible 0.09 vs. sequential 0.29.

• PMI (2018). *Pulse of the Profession.* — 43% miss budget; 52% scope creep; 47% requirements-driven failure.

• PMI. *Toward a Unifying Theory for Compounding/Cumulative Impacts of Project Risks.*

• Prosci. *The Correlation Between Change Management and Project Success.* — success-likelihood evidence (not cost-variance).

• McKinsey/Oxford. *Delivering Large-Scale IT Projects On Time, On Budget, and On Value.* — 45% over budget at $15M+ scale.

• McKinsey. *Cloud-Migration Opportunity.* — 75% over budget, 38% behind schedule.

• RAND (2024). *The Root Causes of Failure for AI Projects.* — ~80% failure rate, ~2× non-AI.

• Gartner (2024). *30% of GenAI projects abandoned after PoC by end-2025.*

• Reference Class Forecasting / DfT–Green Book optimism-bias uplift tables (Edinburgh Tram P50 +40%, P80 +57%).

• Cornish–Fisher expansion. — skew term sign flips at z = 1 (P ≈ 84%); reason it is not used here as a drop-in.

• Log-normal distribution. — mode = $\exp(\mu-\sigma^2)$; quantile = $\exp(\mu+z\cdot\sigma)$.

MyScrumBudget™ Charter Budget Reference · © 2026 William W. Davis, MSPM, PMP · Licensed under GNU GPL v3