⊛ ChatGPT

Finding the "Good Enough" Threshold: Optimizing Risk, Creativity, and Product Decisions

Introduction

Every decision in business, creativity, or engineering comes with a trade-off between striving for perfection and delivering *"good enough"* at the right time. There is an optimal point at which additional effort, time, or risk aversion yields diminishing returns – beyond this point, we incur a **cost of delay or inefficiency** without proportional benefit. This white paper explores that *"good enough" threshold* across three scenarios – cybersecurity risk management, independent music production, and product design – to illustrate how overshooting or undershooting the optimal point can hurt outcomes. We refer to this gap between the optimal outcome and the over-engineered (or over-cautious) approach as the **Inefficiency Delta**. By examining these cases and applying frameworks like Wardley Maps' **Explorers–Villagers–Town Planners (EVTP)** model, we provide guidance for entrepreneurs, creators, and businesses to find the right balance between quality and speed.

Ultimately, this is a guide for decision-makers – from startup founders and musicians to product designers and CISOs – on embracing a *"good enough"* mindset. By calibrating decisions to real-world feedback (be it market response or risk appetite) rather than subjective perfectionism, organizations and individuals can avoid wasted effort, accelerate learning, and achieve better success. In the sections that follow, we delve into each scenario and then extract the common principles and strategies to operate at the optimal point of efficiency and value.

Overshooting Risk Controls vs. Business Appetite (Case 1)

In corporate risk management – particularly cybersecurity and compliance – well-intentioned controls can sometimes overshoot the business's actual risk appetite. **Risk appetite** is defined as the amount of risk an organization is willing to accept in pursuit of its objectives [1] . The board and executives typically set this level to balance opportunity and safety. Trouble arises when security teams impose requirements far below that risk threshold, adding friction that slows the business with little tangible risk reduction in return. This *over-control inefficiency* is one side of the Inefficiency Delta in risk management.

For example, consider a bank that requires onerous documentation and verification to open a basic customer account. Such controls (often driven by compliance or cybersecurity policies) add process overhead. If a new fintech competitor streamlines account creation by only enforcing what is legally necessary (e.g. identity check) and aligns with the bank's true risk tolerance (since a new account with no credit line poses minimal risk), they can onboard customers much faster. The traditional bank's extra verification steps may fall below its risk appetite – meaning the business was willing to accept more risk for the sake of speed – so the additional latency creates **negative value**. It is "security theater" or bureaucracy that **stymies business growth without appreciable security gain**. As Rick Garvin notes in the context of healthcare tech, *"Clampdown too hard on cybersecurity, and you stymie innovation and revenue growth. But if you give innovation free reign, you risk data breaches...that can damage your bottom*

line." [2] . In other words, both extremes – too much control or too much risk – are harmful; the goal is to hit the *sweet spot* of protection that satisfies risk appetite and enables innovation.

Overly conservative security practices incur tangible costs. They delay projects, frustrate product teams, and can cause missed market opportunities. In highly regulated sectors, security "Protectors" (CISOs, compliance teams) sometimes default to **zero-risk tolerance**, preferring *"highly constrained development and data security environments."* They view fast-moving product teams as operating in the "wild west" and reject new ideas on safety grounds [3] . These Protectors often **seem over-controlling**, saying no without considering ways to achieve the business need securely [3] . The result is an organizational drag: product releases are late, revenue is lost, and innovation stalls [4] . In the worst cases, this misalignment between risk controls and business appetite can *"slow down innovation, or even shut it down entirely."* [4]

On the flip side, if risk management undershoots – allowing the business to operate above the defined risk appetite – the enterprise is exposed to threats beyond what leadership finds acceptable. This could mean insufficient security leading to breaches, compliance violations, and catastrophic losses. For instance, a company might push products out with little security testing to speed time-to-market, inadvertently taking on more cyber risk than the board ever intended. Such overshoot of risk appetite is clearly dangerous and falls outside the *optimal zone* of operation.

The key is **calibration**. Cybersecurity and governance teams must calibrate controls to the organization's true risk appetite, which is ultimately set by leadership in pursuit of value creation [1] . When security measures overshoot (i.e. require a risk level even lower than the business needs), they impose an **opportunity cost** – in lost time, productivity, and innovation. This is the Inefficiency Delta in action: resources spent on reducing risk further and further yields no appreciable benefit if you were already below the acceptable risk threshold. As one cybersecurity expert famously put it, *"Compliance is not security. You can be compliant and still be owned."* [5] – meaning you can spend tremendous effort on checkboxes and controls yet remain vulnerable or hampered in creating value.

Finding the Balance: Organizations should aim for the **"good enough" security** that satisfies their risk appetite and no more. That involves risk-based decision making, where low-risk activities face minimal friction, while high-risk ones get stricter oversight. It also requires shifting incentives: risk officers should be rewarded not just for preventing incidents (by saying "no"), but for enabling the business safely. By measuring and communicating risk in business terms (e.g. financial impact, likelihood) and regularly updating risk appetite with the board, teams can focus controls where they matter and eliminate or streamline those that only add drag. The result is an optimal point of "secure enough" – meeting regulatory and threat mitigation needs without wasting resources or time beyond that point.

The Indie Musician's Perfection Trap (Case 2)

For up-and-coming musicians, a common pitfall is delaying the release of new songs until they are "perfect." In this scenario, the *risk* is artistic and reputational – the fear of putting out work that isn't one's absolute best. The analogous **risk appetite** here is actually the audience's tolerance: how much imperfection will listeners accept while still enjoying and sharing the music? Time and again, we see that artists who err on the side of **early and frequent releases** tend to build an audience faster than those who hold back until every detail is flawless. This is because in the early stages, *feedback and exposure outweigh perfection.* Holding a song for months attempting to reach some imagined ideal can mean lost momentum, lost learning, and a lost connection with potential fans.

Consider a musician who has written a decent song – call its quality a 4 out of 10 by the artist's own evolving standards. The audience, however, might have been delighted with quality "3" because it has authenticity or a catchy hook. Releasing the song at quality 4 *right now* would start attracting listeners who find value in it. Perhaps a slightly better studio version (quality 7) could be made with three more months of polishing – but those 3 months of delay are the Inefficiency Delta. **What is the cost of that delay?** During those months, the artist could have released the raw track, gathered listener feedback (Which parts do they like? Did it get shared? Did it "stick" with anyone?), and even used the feedback to create new music. Instead, by chasing a marginal improvement in one song, the artist loses the compounding benefits of continuous creation and audience engagement. Unless the improvement from 4 to 7 brings a *dramatically different audience reaction*, the extra time is essentially wasted effort. In many cases, it isn't – a song that is "good enough" to resonate will resonate nearly as well in a rough form as in a perfected form, especially for a growing artist.

Evidence from the Field: This isn't just theory – successful independent artists often follow the mantra of *"release good music often."* The strategy, as one music industry guide notes, is simple: *"release good music often… It could be every week, every two weeks, or every month."* Frequent releases keep listeners engaged, keep the artist in a creative flow, and even please the algorithms that recommend music [6]. The rise of indie rapper Russ is a case in point. Starting around 2011, he released 11 albums with only modest traction. He then switched to putting out *one new song every week* on SoundCloud for years – an almost relentless pace. Eventually, a couple of those tracks went viral (reaching the Billboard Hot 100), and today Russ enjoys about 15 million monthly listeners on Spotify [7]. Consistency and volume trumped perfection; by giving himself many "at-bats," Russ hit home runs that a perfectionist strategy might never have yielded. Another artist, Nic D, similarly released music biweekly and amassed millions of listeners [7]. The pattern is clear: *a creator in the discovery phase benefits from releasing and iterating rapidly.*

Beyond building audience, there's a psychological benefit to this approach. As LinkedIn founder Reid Hoffman famously said about startups (a principle equally applicable to creatives): *"If you're not embarrassed by the first version of your product, you've launched too late."* [8]. Hoffman emphasizes that speed and learning are more important than polish in early stages. By delaying release, you **delay the feedback loop** that teaches you what your listeners (or customers) actually want. *"If you'd launched sooner, you would have started learning sooner… Instead, you launched too late."* [9]. Musicians who treat each song release as a *one-shot chance at success* often fall into a crushing trap: they spend enormous effort on one big release, only to find it doesn't meet their sky-high expectations. The aftermath is disappointment and burnout – a "sugar crash" after the brief high of the release hype. In contrast, musicians who view each release as an experiment – part of a continuum of growth – stay emotionally resilient. They can respond to what resonates in the last song by incorporating those lessons into the next.

It's important to note that an artist's own definition of "quality" evolves over time. Ironically, as musicians improve their craft, their standards rise; a track they once thought was a 7/10 might only feel like a 4/10 to them a year later. This moving target means that chasing perfection is like chasing the horizon. You never quite arrive, because your perspective keeps changing with experience. That's why external validation (listener enjoyment, shares, feedback) is a more stable measure of success than the artist's internal barometer. If listeners are satisfied with a song that the artist might label imperfect, then it's meeting its purpose. **"Perfect = false,"** as one artist guide bluntly states, urging creators to ditch the Western myth of perfectionism in art [10]. *"Creatively, perfection is also a no-no… Don't try to be perfect… Be ultimate."* [10] – meaning focus on delivering something *real and impactful*, not something flawless.

Finding the Balance: For a musician in the early or "discovery" phase of their career, the optimal point is to release as soon as the music is *good enough to convey its core value to some audience*. This might mean the mix isn't the best, or the vocals could be refined – but if the song itself is catchy or meaningful, those tweaks can be made in a later version or future projects. The *risk appetite* of your audience might be quite high: fans of emerging artists often forgive production imperfections if the music speaks to them. Thus, the artist's focus should shift from "Is this the best I can possibly make it?" to **"Is this effective and authentic for my listeners right now?"**. By aligning output to that threshold, musicians maximize learning and momentum. In practical terms, this could mean setting a release cadence (e.g. one song every 2 weeks or month) and sticking to it, improving production quality gradually but never at the cost of breaking the cadence. As with startups, it's the *first month* or *first year* of iterative growth that matters more than that first day of release [11] . Embracing being a little "embarrassed" by early work is a sign of growth, and it trades short-term pride for long-term improvement and success.

Product Launches and the MVP Mindset (Case 3)

Our third scenario examines a local business launching a new product – say, an artisan beverage or a niche consumer gadget. The entrepreneur faces a classic dilemma: invest heavily upfront in perfect branding, packaging, and inventory for a big splash launch, or go to market quickly with a minimal version and adapt. This is essentially the **MVP (Minimum Viable Product)** question. The Inefficiency Delta here is the difference between a small test launch that could be done now versus a delayed "ideal" launch after months of refinement. Just like in the previous cases, delaying means risking time, money, and missed opportunities if your assumptions about the market are wrong.

Imagine a small beverage company that originally designed its product for young urban professionals. They spend a fortune on a slick logo, premium bottles, and an ad campaign targeting that demographic. After six months, they launch big – only to find out a different segment (say, middle-aged health enthusiasts) actually likes the drink more, and the fancy branding doesn't appeal to them. The company now has sunk costs in branding and inventory tailored to the wrong audience, and pivoting will be costly. This is the danger of **over-investing before finding product-market fit**. In contrast, consider if the company had done a smaller pilot: a basic label, a limited batch, sold at a local market or online, with just enough branding to not look unprofessional. They might have discovered within weeks that an unexpected customer segment is buying most of it, prompting a change in design or marketing messaging. Because they spent little and kept the process quick, they remain agile – with budget and time available to adjust the product or brand for the next round.

This approach is central to lean startup philosophy and modern product management. By **getting a product in front of real customers quickly**, you learn who your real market is and what features or branding actually matter to them. A core concept here is the **Cost of Delay** – the economic impact of not delivering a product or feature sooner [12] . Every extra week or month spent perfecting the product is time that the product is not in the hands of customers generating value (and revenue) or learning. For a new product, the cost of delay is not only lost sales but also lost feedback that could inform a better version. Lean practitioners even quantify Cost of Delay in dollars – e.g. if a product would earn $20,000 per week once launched, then a 10-week delay *literally "costs" $200,000* in missed revenue [13] . While an entrepreneur might not do that math explicitly, the principle holds: *delay is expensive*, often more expensive than a suboptimal initial version.

Another factor is emotional attachment. If a founder pours their heart into one big launch over a year, they become *emotionally invested in that specific vision*. If the market reaction is tepid or suggests changes, it's much harder for them to pivot – they may resist the feedback ("But this is my baby!") or be

reluctant to scrap sunk costs. In contrast, a series of smaller bets – multiple mini-launches or iterations – keeps the founder flexible. Because each version was *just one step* toward the final product, there's less ego bruising in changing course. The founder's identity is not fully tied to any single release, which encourages objective decision-making based on what works.

Finding the Balance: For product companies in the **"villager" stage (to borrow Wardley Map terminology)** – where the concept is known but still evolving to find optimal market fit – the goal is to maximize learning per unit time (and per dollar) rather than to execute a perfect plan. This means aiming for a **"Level 3" design when Level 3 is what customers would accept**, rather than holding out for a "Level 7" polish that customers didn't ask for. If customers start buying (and even recommending) the product at Level 3 quality, that's proof you are in the right ballpark. Incremental improvements can then take it to Level 4, 5, etc., guided by real user input. This iterative scaling is much safer and often faster than an all-in bet on a Level 7 launch that misses the mark.

In practice, techniques such as: - **Minimum Viable Product (MVP):** Develop the smallest feature set that delivers core value to early adopters. This often includes basic branding and packaging – just enough to not turn customers away. - **Pilot Markets or A/B Testing:** Release the product in a limited setting (geographic or online trial) to gauge reaction without a full rollout. - **Staged Design Investments:** Instead of five fancy design concepts upfront, try one or two simple ones in the market test. Save the budget for when you have data on what resonates with customers.

By doing these, a company ensures it doesn't **overshoot the appetite of the market**. Just as a board has a risk appetite, a market has an "innovation appetite" or change tolerance. Customers might be perfectly happy with a simple solution to their problem – a fancier, more expensive solution could even be a turn-off. The optimal point is delivering exactly what meets customer needs and no more. Any bells and whistles beyond that point are features no one asked for (waste), and any quality shortfalls below that point are fatal flaws (risks of under-delivery).

The Common Pattern: Embrace the *Villager* Mindset

Across these scenarios, a common pattern emerges: success comes from operating in the *middle zone* between chaotic exploration and rigid execution. This is akin to what Simon Wardley calls the **"Villagers"** in his EVTP model (Explorers, Villagers, Town Planners). In Wardley's framework, *Explorers* (also known as Pioneers) are the bold experimenters who try radical new ideas with little concern for stability or scale – they create prototypes and often fail fast. *Town Planners* are the opposite end – they industrialize proven ideas, focusing on efficiency, reliability, and scale, often through heavy process and standardization [14] . *Villagers*, however, sit in between: *"They can turn the half-baked thing into something useful for a larger audience. They build trust... listen to customers and turn it profitable."* [15] . Villagers (also called Settlers in some versions of the model) are all about taking an idea that works in principle and making it *good enough* for widespread use, finding the right fit and refining it.

The risk of our three scenarios is essentially **Villagers misbehaving as Town Planners or as Explorers at the wrong time**. If a risk team acts like a Town Planner in a context that still needs experimentation (imposing bureaucratic processes on a nascent project), they create friction that hampers finding the best solution. If a musician or entrepreneur is too much of a Town Planner (striving for mass-market perfection before having a proven product), they over-engineer and stall. Conversely, staying an Explorer too long (never adding discipline or quality focus) means never reaching the audience or market at scale – e.g., a musician who only jams in the garage and never edits or publishes a song, or a startup that keeps pivoting on wild ideas without ever refining one that customers stick with. In all

cases, the **Villager mindset** – iterative development, market feedback, "good enough" execution – is key to bridging the gap between idea and scalable success.

Wardley mapping teaches that different stages of evolution require different approaches: *"How you manage, purchase, finance and build a totally novel concept… is radically different from how you manage and build a highly industrialised commodity. If someone says to you that 'we should use agile everywhere' or 'we should use six sigma everywhere' then they are either inexperienced or… flogging a fad."* [16] [17] . In other words, one size does not fit all. Explorers may use chaotic, rapid experimentation. Town Planners use process optimization and Six Sigma efficiency. But in that middle stage – where our product, song, or project is *partially evolved but not yet a commodity* – we need a hybrid approach: **enough structure to deliver reliably, but enough flexibility to keep iterating**.

The Villager mindset embraces *evolutionary delivery*: release, learn, tweak, repeat. It values customer and user feedback over theoretical perfection. It also recognizes the **law of diminishing returns** – that beyond a certain effort, each extra unit of polish yields negligible improvement in outcome. The Villager knows when a feature is "good enough" to ship or when a control is sufficient for the risk at hand. This is essentially finding the **Good Enough Threshold** – the point at which the product/service meets the necessary quality or safety bar for its context. Any effort beyond that threshold is the *Inefficiency Delta*, which should be avoided or at least questioned critically.

Strategies to Find the Optimal Point

How can practitioners in these diverse fields find and stick to the optimal "good enough" point? Here are some actionable strategies drawn from the scenarios and supporting principles:

- **1. Clearly Define the Acceptance Criteria (Risk or Quality):** Define what "good enough" means for the task. For a security team, this is a **risk appetite statement** – e.g. *"We accept X risk (say, up to a certain dollar impact or probability) in pursuit of this opportunity."* For a product or creative work, define the core elements that *must* be right for the audience. This could be a user story ("the device must safely dispense one cup of juice without leaks") or a musical element ("the song should evoke emotion or make people dance"). Having concrete minimum goals helps avoid gold-plating. Anything beyond satisfying those core criteria might be nice-to-have but not a blocker to release.

- **2. Use Time Boxes and Cost Caps:** Force discipline by limiting how much time or money can be spent before showing results. In agile development, fixed-length sprints ending with a demo enforce the idea that something should be delivered in that timebox, even if it's small. For musicians, setting a release date (or a song-a-week challenge) creates a constraint to finish and share work. These constraints combat the natural tendency to keep tweaking indefinitely.

- **3. Leverage Feedback Early and Often:** As soon as you have something viable, get it in front of real people – users, listeners, or internal testers. Their reactions will tell you if further work is justified or if you're already hitting diminishing returns. In software and product design, this is beta testing or soft launches. In music, this could be sharing demos on social media or performing live. Early feedback is the compass for the next iteration. It prevents you from solving problems that don't exist or polishing facets that nobody cares about. As Reid Hoffman's quote suggests, the feedback loop is golden – every cycle you complete teaches you and increases the eventual quality far more than one big untested push [9] .

- **4. Break Big Bets into Small Bets:** Instead of one massive release or decision, break the journey into phases. This is analogous to **modular design** or micro-releases. For a product, maybe launch in one city before national rollout, or one niche channel before all retail stores. For an album, release a single or an EP (extended play) before a full album, to gauge which songs resonate. Each small bet either validates your path or corrects it with minimal pain. Importantly, celebrate small wins – a modest release that meets its modest goal – to keep motivation high for the next step.

- **5. Align Incentives and Culture with Optimal Risk-Taking:** Within teams, make sure that those tasked with quality control or risk management are *empowered to say "yes"* when appropriate, not only "no." Create a culture where **learning from failure** is praised. When a quick-and-good-enough release doesn't perform well, treat it like a science experiment that yielded valuable data, not a fiasco. This encourages team members (developers, security officers, artists, etc.) to operate within the safe bounds but push right up to them. For instance, a CISO might set a policy that for low-risk projects, security will not hold up deployment – instead they monitor in runtime and adjust. This signals to developers that security's goal is enabling smart risk-taking in line with appetite, not zero risk. In creative teams, encourage open critique and iteration – e.g. songwriters co-writing and iterating on unfinished songs – to normalize the idea that work is expected to evolve in public.

- **6. Use Wardley Maps or Similar Tools:** Map out your initiative's components and determine their evolutionary stage. Is the core product in the "uncharted" stage (Explorer territory), the emerging stage (Villager), or the well-understood utility stage (Town Planner)? Mapping can highlight if you are applying the wrong approach. For example, if your map shows the customer need is well-known and the tech is commodity, maybe you *should* be executing efficiently (Town Planner style) and not reinventing. But if the map shows a novel component critical to user needs, you know that part requires experimentation (Explorer style) and you shouldn't over-bureaucratize it. Wardley's doctrine even advises against blanket methodologies: don't "Agile everything" or "Six Sigma everything" [17] . Instead, use Agile for developing the novel bits and use Six Sigma or automation for the well-understood bits. This ensures each part of the project gets **just enough process** – no more, no less – for its nature.

- **7. Calculate (or at least estimate) the Cost of Delay:** Especially for product launches and features, try to quantify what a delay costs in terms of lost value. As mentioned, if a feature could bring in $N per week, every week of delay "costs" that amount [13] . This can re-frame discussions: a decision to spend extra time polishing is not "free" – it is incurring an opportunity cost. Seeing those numbers can often encourage teams to ship earlier. In creative work, you might quantify differently – e.g. *each month I don't release a song, I lose X potential new fans or Y streams.* While these are estimates, even a rough figure can combat the intuitive bias that extra time always improves things. Often, *speed has value* that rivals quality.

Conclusion

Perfectionism, over-engineering, and ultra-conservatism in risk are all shades of the same fundamental issue: **a failure to define "good enough" and pull the trigger when it's met.** The Inefficiency Delta – that gap where extra effort or caution yields negligible benefit – plagues many endeavors, from business strategy to artistic creation. By recognizing this and actively managing to avoid it, entrepreneurs and organizations can reclaim huge amounts of time, money, and competitive advantage.

The three case studies we explored share a unifying lesson: **find the optimal point and be willing to stop there (at least for now).** In cybersecurity risk management, that means aligning controls with the true risk appetite – protecting the business *adequately* without strangling it. In music, it means releasing work when it's *authentic and decent*, rather than waiting for an ever-elusive perfection – because audiences reward relatability and consistency over polish. In product development, it means launching *when your solution solves the core problem*, even if it's rough around the edges, and iterating with real user input – rather than betting everything on untested assumptions.

This isn't advocacy for sloppy work or reckless risk-taking. Rather, it's about **optimizing for value**. Each scenario has its version of a Goldilocks zone: not too hot, not too cold. The board room's duty is to ensure the company doesn't sail into existential risks (too hot) but also not to instill a fear of all risk that freezes progress (too cold). The artist must avoid releasing off-key ramblings (too hot mess), but also avoid scrapping great songs that just needed a bit of mixing (too cold feet). The product designer shouldn't ship a dangerous or non-functional product (too hot to handle), but also shouldn't spend years in R&D for features users might not care about (too cool to make a splash).

By embracing the mindset of the Villager – pragmatic, user-focused, iterative – we naturally gravitate to that "good enough" threshold. We allow the Explorers in us to experiment and generate options, and the Town Planner in us to systematize proven aspects, but we chiefly act as Villagers who *turn prototypes into products and listen to feedback to drive improvement* [15] . In doing so, we harness the best of both agility and planning.

In sum, finding the "good enough" spot is about **maximizing impact while minimizing wasted effort**. It is a dance between courage and caution. Those who learn this dance – be it a CEO aligning security with strategy, a musician sharing drafts with fans, or a startup founder running lean tests – position themselves to win in the long run. They will outlearn, outpace, and outadapt those mired in analysis paralysis or chasing perfection. The world moves too fast, and opportunities pass too quickly, for any other approach. As the cases here demonstrate, shipping early and adjusting beats waiting in vain for perfection. The optimal decision is the one that creates value now and keeps you in the game to refine later. That is the essence of finding your *"good enough"* threshold and turning the Inefficiency Delta into genuine progress. [2] [18]

(1) Are Your Risk Appetite Statements Compliance Exercises?

https://www.cmswire.com/information-management/are-your-risk-appetite-statements-compliance-exercises/

(2) (3) (4) Is Cybersecurity Risk Management Slowing Down Innovation and Speed to Market?

https://tenmilesquare.com/resources/security/is-cybersecurity-risk-management-slowing-down-innovation-and-speed-to-market/

(5) Cybersecurity vs. Compliance: Why Following the Rules Isn't Enough

https://sanguinesa.com/cybersecurity-vs-compliance-why-following-the-rules-isnt-enough/

(6) (7) The release strategy of successful indie artists | Bandzoogle Blog

https://bandzoogle.com/blog/the-release-strategy-of-successful-indie-artists

(8) (9) (11) (18) If There Aren't Any Typos In This Essay, We Launched Too Late!

https://www.linkedin.com/pulse/arent-any-typos-essay-we-launched-too-late-reid-hoffman

(10) Practical advice for new musicians – The Creative Independent

https://thecreativeindependent.com/guides/advice-for-new-musicians/

(12) (13) What is the Cost of Delay and How to Optimize Your Process

https://businessmap.io/lean-management/value-waste/cost-of-delay

(14) (15) (16) (17) Bits or pieces?: How to organise yourself - the dangerous path to Explorer, Villager and Town Planners

https://blog.gardeviance.org/2023/12/how-to-organise-yourself-dangerous-path.html