

Levelling, Performance, & Promotions

This paper describes a framework for levelling, performance, and promotions for employees in an organization. It proposes 2 main ideas:

- **Levels should be based on a person's ability to contribute** rather than their position in the organizational hierarchy
- **Performance evaluation should be done across people of different levels** using the same level scale instead of just being relative to each level.

The desired outcomes of this framework are:

- Progression and recognition for individual contributors instead of requiring a management role
- Empowering strong performers quickly and preventing entrenchment of underperforming senior staff

Goal

The root question is why should we pay some employees more and other employees less? Putting harder notions of fairness or recognition aside, it ultimately boils down to 3 things:

- Hire good people
- Incentive them to improve
- Keep them working for you

You have to pay some people more because their skills are more in demand. You want to reward people for improving their experience and ability over time. And finally you want to make sure that the good ones stay with you over time.

Practically this means we need to do 2 things:

- Promote people to their stable level of contribution as quickly as possible
- Avoid erroneously promoting someone beyond their ability because promotions are hard to reverse

Problem

Traditionally levelling was synonymous with your job in the organizational hierarchy. You have a pyramid of managers starting from the CEO down through middle management ending at the staff workers. As you performed well you got promoted to a higher position in the hierarchy with a commensurately higher salary. This assumes that the central roles are more critical and you want your most capable people occupying those positions.

The first problem is that this system does not recognize individual contributions and only rewards managerial progression. This creates substantial problems for skilled industries:

- **Good individual contributors may not be good managers**
- Lack of recognition for strong individual contributor drives capable employees to seek opportunities elsewhere, **leaving you with weak individual contributors**
- As weak managers hold onto their positions in the hierarchy, **promotions of strong managers gets blocked**

A famous doctor can easily earn more than his head of department. A good teacher should be rewarded without needing to take on administrative roles. And a star soccer player is paid more than the coach or team manager even though he has to answer to them.

For a system to work with skilled workers, it has to **allow for some level progression independent of managerial progression.**

Levelling

One way to handle this is to have a levelling system with the following properties

Levels should be decoupled from position in the hierarchy

A person's level should be able to progress without moving up managerially. For highly skilled roles, strong individual contributors should easily be compensated more than some managerial roles. A really good software engineer should be able to be paid appropriately without needing to become an engineering manager. Even among managerial roles, a really good team lead should be able to be paid well without becoming a director. Levelling and compensation is based on a person's ability to contribute not on their organizational position.

Positions in the hierarchy have minimum and maximum levels

That being said, it makes sense to place a minimum and maximum levels for a particular roles to reflect the necessary requirements and limits to contribution. For example, a tech lead might have a minimum level as you want someone who is sufficiently capable to properly fill the role. Similarly, you might place a maximum level on executive assistants, as no matter how good they are at the job their contribution will be limited by the role unless they take on more responsibility.

These limits can be narrow or broad depending on the role. So a software engineer role might have an extremely wide range from an associate engineer all the way to a

distinguished engineer that is compensated just as highly as a senior leadership position. On the other hand, a senior director position might have quite a narrow band as you have a high bar for someone to fulfill the responsibilities.

Positions can also have additional allowances associated with them

To entice someone to take on a role with greater responsibility without increasing their level, roles with greater responsibilities can have additional allowances associated with them. So while a senior engineer may not need to be promoted to become a tech lead, he can be enticed to do so with an allowance tied to the role. That way should he decided to revert back to being an individual contributor, he can do so without requiring a level downgrade.

Levels should be based on proven ability

The key point is that levels should not be based on time norms. Time norms are an objective but irrelevant measure. They are “impartial” but do not actually track the performance of an employee. The problem with time norms is that it rewards under performers for staying in your organization longer, while disincentivizing strong performers. Taken together, the outcome is that you have lots of senior but incapable leadership while your most promising new hires leave for places where they are better recognized.

That being said, raw ability is a hard thing to measure. What we should look for instead is proven ability. This means that when hiring someone, you look for evidence of their achievements and accomplishments when deciding how to level them. Number of years of experience itself is a bad measure. But number and scope of projects completed is a good one.

Clear level semantics

The specific levels themselves need to have clear and testable semantics associated with each one. Descriptions like “mastery of multiple disciplines” are not terribly useful. Levelling charts like that often just become longer and longer synonym lists of the word “good”. Testable qualities should be used such as “Able to execute tasks competently without supervision” or “Creates new industry leading standards”

might be good measures for a full software engineer and distinguished engineer respectively. There will always be judgement involved in deciding whether someone meets these benchmarks, but in principle at least the benchmark is a specific behavior or achievement.

Proposed Level Semantics for Engineers

O	L5 Principle	Able to create new industry standards in technology
N		
M		
L	L4 Staff	Able to create novel designs to solve mission critical problems
K		
J	L3 Senior	Able to critically evaluate and decide between a range of viable approaches to implementing a project
I		
H	L2 Core	Able to work independently on a project identifying what needs to be done and executing on it
G		
F	L1 Associate	Able to perform tasks when directed
E		
D		
C	L0 Intern	Shows aptitude but still learning how to perform tasks
B		
A		

Note that even as an individual contributor an employee can be compensated at the same level as senior management positions without taking on managerial responsibilities.

Performance

To achieve this proper levelling, we need to do performance evaluations properly. Traditionally, this is done by grading each person relative to their grade “reaching

expectations", "meeting expectations" or "exceeding expectations" being the usual descriptions with some exceptions for extremely high or low performers. The problem with this system is that it has a "average" bias. A senior software engineer who is not achieving much will probably get an "average" grade unless he messes up quite badly. This is bad because a senior software engineer should be consistently outperforming his lower level colleagues. The question of how well someone is performing becomes obscured by the condition of "for their level".

Instead, **performance evaluation can be done on the same level scale**. So instead of asking whether someone is performing well or poorly "for a level 7", you ask if he is performing at a level 5, 6, 7, 8, 9, or 10 rate.

The use of an absolute scale instead of a relative one for performance evaluation has several benefits:

- **It allows fast recognition of new over performers** so if a fresh grad is performing the same as a senior level 8 it becomes very apparent
- **It avoids entrenchment of old under performers**, so if a senior employee is now actually contributing less than his level he can no longer hide behind an "average" performance rating.

We now have a system where senior contributors should consistently do better than junior ones. This makes incorrectly levelled people obvious as they are now compared across their vocation rather than just within their level. If we then promote/fire people accordingly, levels stop becoming rewards for performance.

Levels are long term performance.

The limit of this system is that global rankings still need to be done across people of sufficiently similar vocations. So while you can compare a principal engineer to an associate engineer, you cannot meaningfully compare them to a UX designer of any level. Among managers, it is hard to compare a team lead to a middle manager. Therefore, you need to still "group" people into meaningful comparison categories. Nonetheless, it still provides a more meaningful evaluation over the relative model.

One logistical advantage of this absolute scale is that it makes calibrations across teams dramatically easier. Instead of having to do calibrations for every level, each manager just stack ranks their own team then only has to compare a few "benchmark" people across teams to ensure that the different stacks are roughly aligned.

Promotions

With this performance evaluation system in place, our promotion system becomes quite simple

- If a person is drastically outperforming his current level
 - Promote
- If a person is slightly outperforming his current level
 - Promote if he has done so for the last few periods
- If a person is performing at his current level
 - We do not need to do anything
- If a person is underperforming his current level
 - Time to discuss performance improvement plans or exiting him

The advantage of this system is that the level someone is hired at becomes irrelevant as they should quickly converge to their level of performance.

CEP considered harmful

CEP is the current estimated potential of an employee. It is what the employees managers estimate a person's "apex" job will be at the end of their career. It is intended to identify junior employees with high potential and promote them accordingly. CEP has several critical problems associated with it:

- It is **extremely hard to meaningfully assess** a person's ability to perform a role several decades from now. It is hard enough to assess a person's current performance, adding several decades in between introduces so much noise that it is **based on little more than "gut feel"** with all the inherent biases it entails.
- While CEP can be adjusted by subsequent managers along a person's career, there is substantial pressure to "stabilize" and deviations have to be substantially justified. As a result **people end up getting pigeon-holed**. People who are designated "high flyers" get promoted quickly despite mediocre performance, and people who are deemed "low potential" can get stuck despite doing well at their job. While both these scenarios can be worked around by a substantially motivated manager, there is a very strong status-quo bias that cannot be discounted.

- There is a self-fulfilling prophecy. It **encourages superstition around factors of success rather than actual success**. Subtle “common sense” around manners of speaking, showmanship, and even body language feed into a person’s CEP and therefore their promotion rate. People who are successful using drastically different strategies are penalized in favour of those who more closely resemble what the “image of success” looks like.

With an absolute performance evaluation scale and the simple promotion criteria, **CEP can instead be replaced by a person’s current actual performance (CAP)**. So if a employee is indeed outperforming his level, we can much more accurate assess that and promote him accordingly. The incentives to perform well become much more well defined and we avoid the current exercise of trying to predict the future from the equivalent of personality tea leaves.

CEP can work in cases where a person’s capability is clear from the start. However, **almost all cases where CEP would have gotten it right, using actual performance would have done just as well**. The only case where CEP is useful is to “smooth over” necessary but drastically different low-level jobs for people who might make good senior managers but poor workers. However, the potential for “abusing” such a justification for is substantial. These should be the exception rather than the rule, and require quite **explicit recognition that a person is being promoted despite his mediocre performance** at the low level.

Push Promotions & Equal Compensation

Instead of waiting for people to ask for promotions, you should actively try to promote people as much as possible. If you require people to push for their own promotions, you end up promoting people who spend a lot of time self-promoting. You want to reward people for doing well instead of petitioning for promotions.

Similarly, you want to keep compensation within each level for each role the same. Avoid creating individual contracts and exceptions for compensation. You do not want to create an incentive where people are rewarded for spending time getting competing job offers. People should be levelled appropriately and not have to worry about fighting for their own compensation.

A Note on Bonuses

Promotions are used to bring someone to their appropriate level of contribution. Bonuses on the other hand are used to reward temporary level outperformance.

Bonuses should not become routine by definition. They become expected and therefore useless.

Bonuses should be used to reward temporary level outperformance. So a senior engineer doing well at his level should not get a bonus. But a software engineer who performed at a senior level should until he is promoted himself.

Ideally bonuses would be paid out as close as possible to the period of overperformance. The limits to this are that it takes some time to evaluate the impact of a person's work, and performance evaluations are administratively expensive. It is therefore necessary that there is a longer delayed cycle of bonus evaluation.

The problem with standard bonuses

Expected bonuses are just delayed compensation. Rather than simply paying a worker in full each month, a portion of the total compensation is withheld until the end of the year. Even after the bonus period of a year has passed, some organizations delay the actual payout of the bonuses even further for a few months after that.

The total amount to be paid to the worker is already agreed in advance. So any expected bonus necessarily comes from a reduction in monthly salary. Instead of delaying till the end of the year to pay someone, that money should be paid out earlier when the work was done.

This practice is immoral. A common argument in favour of it is that it promotes retention since workers have the incentive to stay until bonuses are paid out. Work completed means money owed. And money owed should be paid out as soon as possible. An organization delaying bonuses is knowingly coercing further work from their employees by not fulfilling their debts. This practice of withholding wages is in fact in violation of many countries' labour laws.

The immorality of this practice is made all the more clear with some organizations actually correctly paying out a worker's bonus in advance over the normal wage cycle, but then requiring the worker to pay back the money if he quits before the "actual" bonus date.

Morality aside, this practice is actually counter-productive. Retaining employees through coercion means selectively retaining low morale and unmotivated staff. These are the staff that ideally should be exited as soon as possible. While in the very short term you may gain some productivity, even in the medium term you actually want to be searching for replacement candidates as soon as possible.

Furthermore, it creates a spike in resignations after bonus periods. Instead of a smooth turnover of employees throughout the year, it places a heavy burden on specific periods which could result in more acute understaffing, administrative burden for recruiting, or even lower hiring standards due to the urgency.

Summary

- Levels should be based on your ability to contribute and are distinct from your position in the hierarchy
- Performance evaluation should use the same level scale to make the link between levelling and performance clear
- Promotions should be done quickly but judiciously

Everything after this line are legacy documents. For the most updated career schemas, look for the respective files on google drive.

=====(Leonard)=====

Potential grading structure for PMs:

Area	Description	Evaluated by	Points allocated
Persuasion	<ul style="list-style-type: none"> • Ability to navigate political roadblocks to roll out a product • Ability to secure funding and buy-in • Presentation and communication skills • Marketing skills 	<ul style="list-style-type: none"> • Director's observation • Teammates sitting in on meetings and presentations 	1 to 7
Product	<ul style="list-style-type: none"> • Design thinking skills and knowledge of UX frameworks • Empathy, and understanding of what end users need 	<ul style="list-style-type: none"> • UX Designer you're working with • Fellow PMs 	1 to 7
Engineering	<ul style="list-style-type: none"> • Infrastructure (DevOps) knowledge • Ability to write scripts for analytics • Ability to write code, and work on PRs with engineers 	<ul style="list-style-type: none"> • Engineers you're working with 	1 to 7
Teamwork	<ul style="list-style-type: none"> • Ability to be organised and structured in managing tasks • Ability to keep team motivated and focused 	<ul style="list-style-type: none"> • Team feedback 	1 to 7
Outcomes		<ul style="list-style-type: none"> • Citizen satisfaction • Cost savings • Man hours saved • Agencies impacted <p>Weighted by % involvement in project</p>	1 to 14

Grades A to O based on the point system from 1 to 42

Levels

Intern - Interest in building products and solving practical user problems

- Intern 1 - Entry point for students who are interested to be a PM
- Intern 2
- Intern 3

Associate - Entry level individuals who have the qualities and potential to be a product manager

- Associate 1 - Entry point for most fresh graduates
- Associate 2

Intermediate - Competent individual contributors

- Intermediate 1
- Intermediate 2
- Intermediate 3
- Intermediate 4
- Intermediate 5 - Resting level for most product managers

Senior - Fully proficient individual contributors with the experience and knowledge to build products

- Senior 1
- Senior 2
- Senior 3

Split between individual contributors and managers. High level individuals can operate independently to build successful products while group product managers would typically be in charge of a segment of products and lead PMs within that segment

Manager - Fully proficient product managers who are competent people managers

- Manager 1
- Manager 2
- Manager 3

Principal - Fully proficient product managers who responsible for managing the product strategy and development for a line of business

- Principal 1
- Principal 2

Product Manager - Individual Contributor Track

A PM should excel in three core things: a) product insight, b) ability to work with a team of designers and engineers to transform this insight into a usable product and c) persuasion skills to get users, and resourcefulness to get policy and funding support to deploy the product. A fourth bonus dimension is tech know-how which is a force multiplier for the above three areas - a) with analytics skills a PM can generate data-driven product insight, b) with knowledge of engineering task complexity a PM can better evaluate time needed per task and prioritise tasks correctly, and c) a PM can better persuade stakeholders by explaining the most technical details in laymen language.

Associate Product Manager

Associates are **entry level PMs**. They should have the product sense to empathise with a user and identify user problems, but might rely on a more senior PM to define big product opportunities. They should have good teamwork, communication skills and some tech know-how. Concretely this means:

Design (product features), tech, analytical thinking, and project management
(teamwork/communication)

- **Product:**
 - Enumerate logical reasons why products are good/bad- able to find product market fit, understanding of product design
 - Execute an established product vision by defining relevant features
 - Write requirements understandable and actionable by engineers and designers
 - Track key metrics and articulate rationale behind the chosen metrics
- **Project management:**
 - Articulate how you would go about organising the team
 - Identify attendees, set agenda, and run meetings
 - Escalate when there's an issue
- **Analytical thinking:**
 - Break down a problem statement into actionable items

- **Tech know-how:**

- Have knowledge of how tech systems work and if not, will have the initiative to read up, e.g. know how DNS or RSA works
- Able to estimate engineering effort needed with some assistance from engineers and/or more senior PMs

Intermediate Product Manager

Intermediate Product Managers are **competent and independent PMs**. They should have strong initiative, and be able to independently define big product opportunities

They should have the product sense to empathise with a user and identify user problems, but might rely on a more senior PM to define big product opportunities. They should have good teamwork, communication skills and some tech know-how. Concretely this means:

They plan a product's roadmap, define clear milestones and KPIs for the product. They have the communications skills influence and align the team and stakeholders towards the products' goal. They lead team sizes between 3-5 members as well as work with cross departmental groups.

- **Product:**

- Define product-specific measure of success and identify relevant metrics
- (APM) Track measures of success, (PM) Prioritise measures of success and know which to ignore
- Design and define product features through talking to users or studying the data (JD like)
- Prioritise features while balancing multiple considerations such as end user utility, feasibility (JD like)
- Translate feature into requirements understandable by engineers
- Come up with product hypothesis and verify them through user research (JD like)
- Quick at evaluating tradeoffs to make actionable decisions, even under pressure (Correct short and long term measures to remedy and prevent future crisis)
- (PM - be able to pinpoint core advantages and disadvantages)

- **Teamwork:**

- Able to persuade engineers and designers to want to join the project and subscribe to the product vision

- Impart a sense of ownership and product purpose to team
- Introduce/Adapt team management tools and processes to improve the team's productivity
- **Persuasion:**
 - (Identify and)/(Map out and prioritise stakeholder engagement, then) influence stakeholders prerequisite to product release (e.g. people, policy, funding, legal, ..)
 - Able to independently convey product to all audiences, including to a large audience or a high level official, e.g. PS (adapt messaging to audience and engage audience, ideas/benefits) (to the press)
 - Able to run efficient meetings independently to audiences of all levels (including at the PS level and above)
- **Tech know-how:** (flesh out more which tech know hows)
 - Have knowledge of how tech systems work and if not, will have the initiative to read up, e.g. know how DNS or RSA works
 - Independently estimate engineering effort needed
- **Traits:**
 - Ability to recognise failures and correct them
 - Independence
 - Initiative
 -

Senior Product Manager

Senior Product Managers are experienced and proficient product managers. They are aware and able to implement product management processes and frameworks to the team they are placed in. They should be able to work on complex products.

A senior product manager should possess as much of the following qualities in addition to the previous:

- Take point on complex, cross team projects/products
- Identify new opportunities and find product fit
- Identify edge cases and long term use cases in product requirements
- Ability to mentor Associate or PMs
- Build product awareness cross teams
- Influence adjacent divisions to support product vision
- Plan and run larger scale product marketing

Technical

- Technical depth and ability to pick up newer technical concepts

Principal Product Manager

Principal product managers are full proficient individuals who prefer to be on the individual path instead of going on the managerial path. Principal product managers are effectively more experienced senior managers who are recognised.

Product Manager - Managerial Track

Group Product Manager

Group product managers are senior product managers who have chosen to go on the managerial track to lead people. The group product manager would be in charge of the roadmap and KPIs of a segment/vertical of products. They would be leading a team of product managers whom would be individually in charge of a sub segment of the product. Group product manager would provide the high level vision and roadmap as well as guide and track the individual product managers reporting to him/her.

Product Management

- Align products with company objectives
- Identify competitive landscape and features that would future proof product
- Set and drive segment measurements and KPIs
- Formalize processes and frameworks and adoption of best practices
- Manage and mentor product managers

Communication

- Collaborate across departments/division to drive success of product
- Plan cross product/segment marketing

References:

- <http://www.sachinrekhi.com/product-management-career-ladders-at-8-top-technology-firms>
- <https://productsthatcount.com/resources/video/stripe-product-lead-on-managing-product-managers/>
- <https://www.teamblind.com/article/Product-manager-levels-at-Google-vTJLT2uH>
- <https://www.levels.fyi/?compare=Google,Amazon,Microsoft&track=Product%20Manager>
- <https://medium.com/pminsider/product-manager-skills-by-seniority-level-a-deep-breakdown-cd0690f76d10>

ALL Product Manager

- Product:
 - Define product-specific measure of success and identify relevant metrics
 - Seeks/Guidance for big and long-term product opportunities (Identify cross products synergies within org)
 - Able to design and define product features through talking to users or studying the data
 - Prioritise features based on end user utility
 - Translate feature into requirements understandable by engineers
 - Come up with product hypothesis and verify them through user research
 - Quick at evaluating tradeoffs to make actionable decisions, even under pressure (Correct short and long term measures to remedy and prevent future crisis)
- Teamwork:
 - Able to persuade engineers and designers to want to join the project and subscribe to the product vision
 - Impart a sense of ownership and product purpose to team
 - Introduce/Adapt team management tools and processes to improve the team's productivity
- Persuasion:
 - (Identify and)/(Map out and prioritise stakeholder engagement, then) influence stakeholders prerequisite to product release (e.g. people, policy, funding, legal, ..)
 - Able to independently convey product to all audiences, including to a large audience or a high level official, e.g. PS (adapt messaging to audience and engage audience, ideas/benefits)
 - Able to set and drive the agenda to audiences of all levels (including at the PS level and above)
- Tech know-how:
 - Have knowledge of how tech systems work and if not, will have the initiative to read up, e.g. know how DNS or RSA works
 - Independently estimate engineering effort needed
- Traits:
 - Ability to recognise failures and correct from them
 - Independence
 - Initiative
 - Relentlessness / Stubbornness

[May 9th version]

Associate

Associate Product Managers are **entry level product managers**. They have some expertise and demonstrated potential in four key areas of product, project management, analytical thinking and tech know-how. They are able to work with engineers and designers to execute on an established product vision with some guidance. Concretely this means they are able to:

- **Product:**
 - Enumerate logical reasons why products are good/bad
 - Execute an established product vision by defining relevant features
 - Identify suitable target audience and accurately measure utility to target audience
 - Define and track a range of suitable success metrics
- **Project management:**
 - Synthesise team information such as features, tasks, engagements, etc. in an organised and shareable format
 - Write requirements understandable and actionable by engineers and designers
 - Independently organise meetings: identify attendees, set agenda, and run meetings
 - Identify project roadblocks and weigh resolution methods
 - Confidently and coherently present to all audiences, large/small, staff/management
- **Analytical thinking:**
 - Answer product questions in a manner backed up with facts and data, which involves possessing data fluency:
 - Know what data is available: frontend and backend events, user data
 - Know how to query, analyse, visualise and communicate data
 - Debug an issue by pattern matching with available resources on hand such as data and bug reports
- **Tech know-how:**
 - Estimate engineering effort needed, with some assistance from engineers and/or more senior PMs
 - Have an established baseline of technical understanding with willingness to invest more over time. May not master technical topics but should be able to ask intelligent questions off them. Topics may include:

- Failover, high Availability, redundancy
- Internet architecture concepts: Domain -> Registrar -> DNS -> CDN -> Servers
- Data transfer concepts: SFTP, API, REST/SOAP
- Databases and backups: SQL, NoSQL
- Basics of widely used encryption such as SSL/TLS, symmetric and asymmetric
- Error logging and alarms
- Understanding of tradeoffs between modern frontend/backend web frameworks

Intermediate

Intermediate Product Managers are **independent product managers**. They have demonstrated competence in the four key areas outlined above in addition to what associates have, and can independently run a project from idea to production. They can envision big product direction and opportunities and chart a path to get there. Concretely this means they are able to:

- **Product:**
 - Not just explain why products are good/bad but also suggest various ways of how products can deliver more value
 - Create effective value propositions, and envision big product direction and opportunities
 - Track current and key metrics, and clearly explain why they are key to a particular product at a particular product lifecycle
 - Confidently make good product design calls without engaging a designer
- **Project management:**
 - Run efficient meetings with clear objectives, all decision makers represented, and concrete action items
 - Switch between engineering, design and layman vocabulary depending on audience
 - Adept at prioritising tasks by:
 - Evaluate feasibility, impact and urgency of features
 - Balance offense vs defense features (new features vs operations/technical debt)
 - Independently write emails and papers that convince and persuade relevant stakeholders such as in areas of legal, funding and policy
- **Analytical thinking:**
 - Independently break down complex problems into actionable and tractable sub-problems
 - Debug an issue by identifying multiple possible reasons drawing from not just resources on hand but also past experience and intuition
- **Tech know-how:**
 - Independently and accurately estimate engineering effort, which stems from familiarity with project code bases
 - Confidently reply technical questions without consulting an engineer

Senior/Principal

To be reviewed

Senior/Principal Product Managers are **veteran product managers**. They should already possess skills an intermediate has, but at a higher level of competency. They are not constrained by current opportunities and can envision long-term product opportunities. They are also champions at execution, with strong relationship management skills, a knack for simplifying implementation, and determined to overcome obstacles and get things done. Concretely this means they are able to:

- **Product:**
 - Critically evaluate a range of possible short and long-term product directions, without being constrained by current resources
 - Identify opportunities that arise from new technologies
 - Simplify features that deliver most of the value at a small fraction of implementation effort
 - Forecast benefits of features by applying past experience and leveraging suitable benchmarks, and iteratively update forecasts after release
 - Instead of deriving insights from data, develop intuition then validate it with data
- **Project management:**
 - Showcase outstanding relationship management skills:
 - Inspire and motivate team to reach their potential
 - Inspire end users to have faith in the product when things temporarily go south, such that they trust and cooperate rather than doubt and complain
 - Build authentic and trustworthy relationships with stakeholders that encourages them to take the initiative to support the product
 - Possess deep understanding of how the organisation and related organisations work and have built social capital
 - Make a case that is virtually impossible to refute or ignore by not just using data and past experience, but also biases, beliefs and levers
 - Execute without constraints on oneself - actively pick up skills and patch gaps in the team
 - Preemptively identify potential project roadblocks and potential resolution paths
- **Analytical thinking:**
 - Anticipate implementation challenges (technical, design, bureaucratic) with each feature

- Figure creative ways of getting from point A to B despite there not being precedence
- Anticipate issues that might come up with production-grade products and devise preventive measures
- **Tech know-how:**
 - Intimately understand technical tradeoffs at multiple levels such as hardware, programming languages, system architectures and hosting environments
 - Able to recognise between great and good design, and can articulate directions to get from good to great