

# Whatfix

## Engineering Leveling System

### Update History

| Version | Date             | Updated by  | Remarks                       |
|---------|------------------|-------------|-------------------------------|
| 1.0     | December 5, 2022 | Vara Kumar  | Document Drafted & Circulated |
| 1.1     | January 24, 2023 | Gaurav Bali | Whatfix Branding Updated      |

Whatfix is building a new category of software, That means a lot of innovation has to happen. We want to build an environment where engineers can go deeper, iterate faster, make thoughtful design/architecture decisions, own the system end to end (requirement to monitoring).

We have built a set of guiding principles around which we have created an Engineering leveling system. Leveling system shows the growth path for our engineers.

### **Scope/Impact & Behaviors.**

Levels are described by the scope covered by the engineer & impact made. Along with the behaviors that are expected to be exhibited, Behaviors are grouped into categories like ownership, technical execution, team building, Customer First mindset.

### **You can be technical & grow.**

Movement to management is not considered to be a promotion. It is a different role. An engineer can choose to be a hands-on coder & solve deeper technical problems & grow as a distinguished engineer..

### **Engineering management is technical.**

Managers can not make right decisions without understanding technology. It is not enough for a manager to be a people & process person. Managers need to know enough to be able to contribute in design or code reviews, Be able to communicate with other teams.

### **Prove first before level up.**

Level up happens only after an Engineer is able to prove that he/she has already operated at the new scope & demonstrated the impact expected. Level up does not add additional responsibilities, Level up happens only after additional responsibilities are already handled. It is not mandatory that each & every behavior is exhibited but needs to prove that all categories are covered.

### **All engineering work streams are treated equally.**

Levels are the same across different streams of engineering, No separate expectations per stream (Frontend, Backend, Data, Devops, SRE, QA etc). The criteria of level up is exactly the

same, Engineers need to handle bigger scope, make larger impact & show right behaviors to grow irrespective of their work stream.

### **Years of experience is not mandatory to level up.**

Years of experience is required to master anything, It is not enough to be able to do it once, It has to be repeatable. For instance, You can estimate accurately once but that is not sufficient, there has to be consistency & uniformity in being able to estimate well. Minimum 1 year of experience in the current role is required to be considered to level up apart from that there is no minimum years of experience requirement for any level. The years of experience is only a guidance to set expectations.

### **Our interests change.**

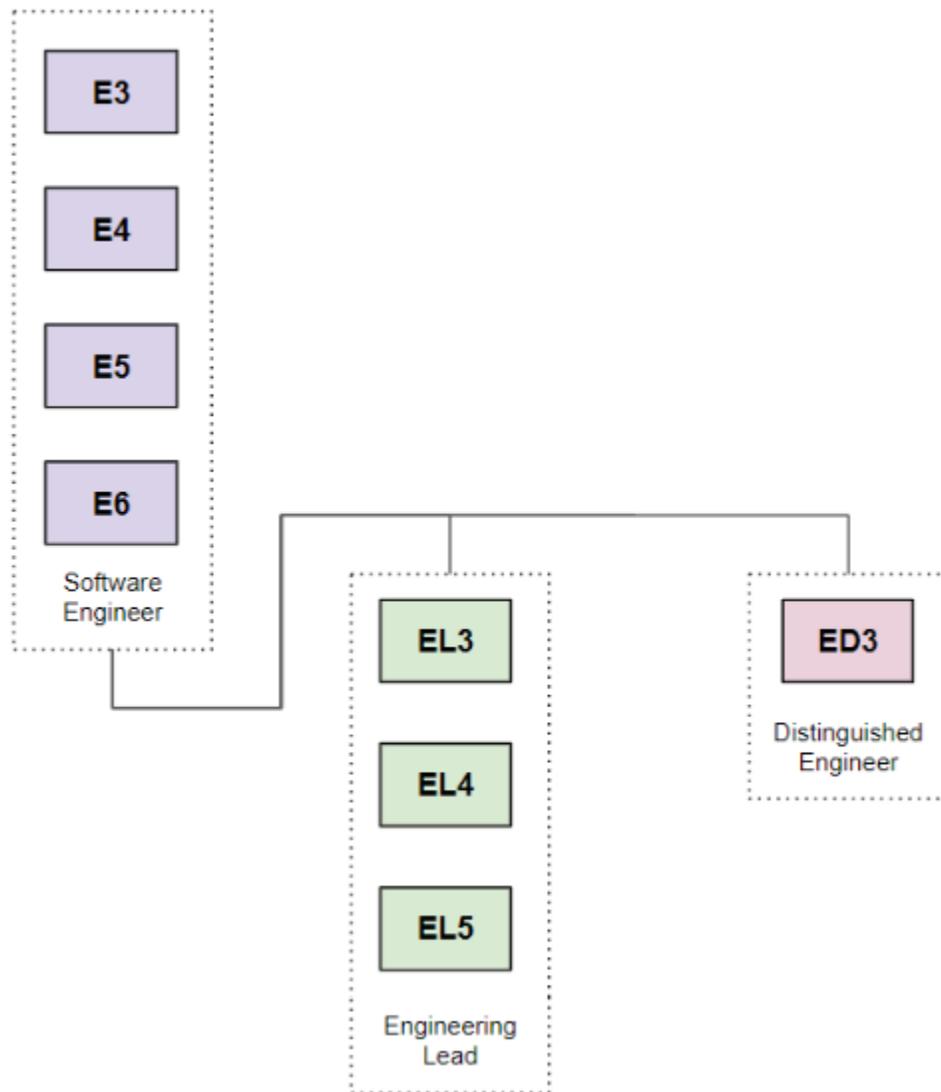
It is not compulsory to stick to managerial or architectural or distinguished paths, You can shift between these areas. A distinguished engineer might develop interest to pursue engineering leadership. The levels can not be compared between engineering paths or with other departments (For instance, There is no comparison between ED3 & EA3, Both are two different expectations).

### **Levels are additive.**

Every level is additive of all previous levels. Responsibilities of all previous levels expected when you are at a level.

*Thankful to Square, Khan Academy & Fog Creek for sharing their leveling systems. We adopted most of Square's philosophy of leveling.*

## Levels



Expectations from each level has been detailed out here:

- Software Engineer
  - [E3](#)
  - [E4](#)
  - [E5](#)
  - [E6](#)
- Engineering Lead

- [EL3](#)
- [EL4](#)
- [EL5](#)
- Distinguished Engineer
  - [ED3](#)

## E3

Typical experience: A new engineer with potential. Typically a person's first or second full time engineering job.

Scope & Impact: Writes, tests, and documents code according to Engineering standards and practices.

### Technical Execution

- Debugs and fixes issues in development, test, and production
- Participates in software design for features and bug fixes under direct supervision.
- Participates in code reviews.

### Collaboration

- Asks for help as appropriate and learns from it.
- Contributes to team activities and collaborates within the team to solve problems
- Openly share their opinions. Is open to changing their perspective and plans based on others input

## E4

### Scope & Impact:

- Designs features with guidance from more experienced engineers.
- Develops, tests, ships, and maintains features independently.

### Technical Execution

- Improves the development experience by enhancing development tools, test coverage, and/or code structure.
- Implements & documents code that is clear, concise, tested, and easily understood by others.

### Ownership

- Is responsible for the entire lifecycle of their code: development, test, production, support and subsequent fixes and improvements
- Serves as on-call first responder for software their team owns.

### Collaboration

- Performs code reviews that follow the standards and practices and that are recognized by their team as helpful.
- Align on the design & code of other teams on the areas where you will need to touch their codebase.

## E5

### Scope & Impact:

- Independently responsible for the entire lifecycle of features including design, development, and deployment.

### Technical Execution

- Improves code structure and architecture in service of testability and maintainability.
- Writes, co-writes, and reviews design documentation.
- Is highly proficient in one or more technical areas

### Ownership

- [Shared] Prioritizes and values unowned or undesirable work that enables the team to move faster

### Collaboration

- [Shared] Works with their team and adjacent teams to solve problems. Escalates problems that have wider scope.
- [Shared] Guide other teams in terms of making design choices who are going to touch the code owned by this team.

### Team Building

- Assists and teaches other engineers on an individual basis.
- Participates in the hiring process (meeting candidates, attending recruiting events, presenting externally, or writing for Engineering blogs).
- Conducts engineering interviews.

## E6

### Scope & Impact:

- Leads or significantly contributes to medium-to-large feature releases; usually multi-person projects that cross engineering team boundaries.
- Is a clear expert in one or more technical areas.

### Technical Execution

- Is recognized for high quality, impactful technical contributions.
- Participates in the design review process, seeking and providing constructive criticism
- Demonstrates ability to succeed in a wide range of complex situations across multiple axes: e.g. scale, uncertainty, interconnectedness
- [Shared] Monitors and adjusts team pace to instill urgency for success but protect from burnout

### Ownership

- Estimates and manages project timelines and risk.

### Collaboration

- Builds relationships with stakeholders including customers, product managers, cross-functional partners, and external partners.
- [Shared] Contributes to defining strategic direction, planning the roadmap, escalating issues, and synthesizing feedback to ensure team success
- **[Shared]** Ensures that knowledge is shared amongst the team and does not position themselves or others as a single point of failure.

### Team Building

- Mentors multiple engineers.
- [Shared] Works through others. Lifts the skills and expertise of those around them.
- [Shared] Recruiting participation increasingly becomes filling open engineering headcount across the company, not just for positions on the immediate team.

## ED3

### Scope & Impact:

- Owned & grew strategic or complex intellectual property of the company. The intellectual property has to be a major differentiating factor for the company in the market.
- Sets technology vision of the intellectual property & new products that company can venture into.

### Technical Execution

- Handson & Solved most complex technology problems that made a large impact on the company.
- Demonstrates ability to quickly analyze and resolve tactical issues across a wide variety of areas.
- Has deep technical domain knowledge of the company. This can cover apps, services, systems, or frameworks.
- Recognized as an expert not only by peers and by the industry.
- Published several patents & papers around the intellectual property.

### Engineering Strategy

- Defines intellectual property strategy and executes to improve products, infrastructure, processes, or company.
- [Shared] Influences senior leaders on engineering strategy

### Collaboration

- Understand & influence product strategy short & long term by being in touch with customers & understanding competition.
- [Shared] Collaborates across teams and disciplines to solve problems and resolve technical debates.

### Team Building

- Publicly known for expertise in a technical area, Speaker at events.
- [Shared] Mentors other engineers or managers on strategy, collaboration, influence, execution, and other aspects of leadership

- [Shared] Recruiting participation increasingly becomes sourcing, selling, and closing offers for managers, senior ICs, and/or tech leads.

## EL3

### Scope & Impact:

- You are owning everything engineering of a product area that is part of a product or platform.

### Leadership

- You are directly managing engineers and optionally managing more engineers through another EL. You have around 7 reportees.
- Communicates strategy, rationale, and progress within the team and to external stakeholders.
- Empowers team to own the product area end to end including prioritization, development, deployment, monitoring, and supporting customers.
- You have a pulse on the infrastructure cost incurred by the product area & be able to control it effectively.
- Rallying team to adhere to 'Definition of Done', Not just as a tick mark but sincerely making use of that construct to improve the quality of product.

### Technical Excellence

- Hands on by picking up at least a development task every sprint, It can be as small as a SUCC issue.
- Upholds a high technical bar while making pragmatic tradeoffs. Able to represent and be accountable for the technical output of the team including design/architecture adherence, through personal expertise.
- Identify and execute features that require spikes & resolve technology bottlenecks before the feature is taken up for implementation.
- Accountable for code quality, design quality, performance, and scalability of the product area.
- You drive reusability by prioritizing designs that help build reusable libraries & platforms.

### Prioritization

- Plans sprint roadmap based on your technology understanding of what should go soon.
- Estimates & help the team estimate the sprint tasks.

## Performance & Team building

- Responsible for coaching, development, and career management: setting performance expectations, career planning, performance reviews, managing low performers, compensation reviews, promotions, and terminations. You take help from leaders & HR wherever necessary.
- Resolves interpersonal and technical conflict within and outside the team.
- Recruiting participation increasingly becomes filling open engineering headcount across the company, not just for positions on the immediate team.
- You cultivate a team culture where near-real-time appreciations are publicly expressed.

## Collaboration

- Guide other teams in terms of making design choices who are going to touch the code owned by this team.
- Collaborates with stakeholders including customers, product managers, and others in development and execution of the team's roadmap.
- Ensures that knowledge is shared amongst the team and does not position themselves or others as a single point of failure.

## EL4

### Scope & Impact:

- You are owning everything engineering of a product area that is part of a product or platform. You built a highly proficient technical team that is able to handle complex problem statements.

### Leadership

- You are a team leader who has a high technical bar & be able to build and drive a highly proficient technical team. You are directly managing engineers and optionally managing more engineers through another EL. You have around 7 reportees.
- You improved the quality of the system, not just bug reduction, lesser technical debt, good control on non-performance aspects etc.
- You are proactively on top of customer concerns, Someone doesn't have to pull you in, You are proactively looking out & reaching for opportunities where you can add value to customers.
- You have a good understanding of customer expectations & contribute meaningfully to roadmap discussions.
- You are delivering more scope with lesser or with the same amount of team from time to time. You are not doing this by getting people to slog but by improving the technical bar of the team, removing blockers that take up load on engineering.

### Technical Excellence

- Recognized as an expert by the team.
- You can solve some deep technical problems that the team is facing. You do not need to rely on architects or engineers to make every technical decision.
- You are driving the application of generative AI in the product areas that you own.

### Prioritization

- Prioritizes enough automation that improves velocity of team and removes manual work.
- You plan for innovation by investing an appropriate amount of time from the engineers.

### Collaboration

- You groom sales engineers, solution engineers, technical support teams to be able to use the product area effectively.

## EL5

### Scope & Impact:

- Manages one or more product lines (or platform) that directly impacts the company's success.

### Leadership

- Your impact is measured on direct business metrics such as revenue, win ratio, gross margin, retention etc.
- You are responsible for budgeting the product/platform that you are responsible for. You are keeping a right tab on budgets.
- You are proactively on top of customer escalations & important opportunities. You don't need an invite from someone to get involved.

### Prioritization

- You are setting priorities for the product/platform by collaborating with the rest of the organization & having a great pulse of customers.
- You will get teams to stop building features & focus on quality if the situation demands.

### Performance & Team building

- You made sure that the teams are structured in a way that they have full clarity on responsibilities & the overlaps are not there. For instance, No success issue needed to be routed through multiple teams.
- You have a pulse on the team on the ground with your direct relationships.
- You made sure that everyone of your team members has clarity on expectations, what it takes for them to move to the next level in their career.
- Brought in enough transparency into the compensation that there is parity across your teams.
- You are proactive in identifying the right skills to augment your team & drive hiring to get those before it becomes a burning need.
- You make sure that retention of team members is topnotch.
- You make sure that time to a hiring decision is improving with time.

### Collaboration

- Build trust from customer facing teams & ensure that their voice is heard.

## Frequently Asked Questions (FAQs)

**1) Is there a direct mapping which exists between different levels (eg. EL3 to EA3)?**

There is no direct mapping available between different paths within Engineering. All movements between paths will be based on the fit with the leveling criteria & availability of such a role (Like regular IJP program).

**2) How will we map employees in case they want to move to a different career role / Business Unit (eg. Engineering role to Product or Design role)**

All such movements will be driven by hiring requirements and individual's fit into the aspired role. Hiring manager needs to evaluate, review & decide the levels.

**3) How often do we revisit the criteria of levels?**

The document is expected to be a living document & all of us are expected to comment/recommend changes to it.