

Software Engineering

Status: Ready

Author: Aike Sommer

Contributors: Ferenc Szeli, Hans Aerts, Roland van Venrooy, Christian Rexwinkel, Rogier Mink

Aim of the Document

Most “doers” in tech organizations – those employees that spend their time building the products that solve our customers problems – are software engineers. To build better products sooner we need to extend the accountability of engineers to include understanding and solving the problems we need to solve. Because software engineers are best suited to assess what is most feasible, this leads to better solutions – sometimes tremendously better. By reducing “communication hops” and the need for multiple functions to get involved in a given deliverable we expect to reduce wait time and information lost in communication and to become more effective. This is an empowering way of working for software engineers that leads to more ownership, higher engagement, and increased job satisfaction.

This document outlines how *accountable behavior* and *corresponding impact* can grow throughout an engineering career and how we intend to incentivize behaviors that increase them both. A deeper look at what this means for other related functions – the “enabling functions”¹ – is not part of the scope of this document. A role description and career ladder outline for engineering managers (EM) can be the Engineering Management – final document.

Scope

Currently there are 4 career paths under the job family *Product Engineering*: Software Engineering, Architecture, Test Engineering, and Technical Writing. As part of the refinement of career ladders the job families Test Engineering and part of Architecture will be merged into the Software Engineering career ladder. Test Engineers that predominantly create automated tests – usually with a similar toolset as Software Engineers – will be transitioned into the Software Engineering career ladder as well. This document is about the individual contributor (IC) track on this newly merged career ladder. The ladders Technical Writing and Test Engineering, when it is not clearly a software engineering function, are out of scope.

The Software Engineer

The Software Engineer both solves customer problems and ensures sustained health of the team’s product and systems – ultimately by writing and reviewing code, operating it in production, and everything in between. She strives to understand the customers’ needs, the product, the problems to be solved, and the strategic direction of her team. Her priorities and the impact of her efforts² help the company, the team and herself – in this order. While the toolset she uses is centered

¹ The purpose of most other functions in our product teams should clearly be understood as “enabling functions”. While sounding like a minor detail, it is an important mindset shift to fully seeing PMs, EMs and other roles as “only” enabling software engineers to build better products faster.

² The nature of these efforts can vary a lot – the default is that the software engineer is solving the problems the team is asked to solve and that this is “moving the needle” in a meaningful way.

around developing and operating code, she sees her responsibility as primarily around solving customer problems and delivering tangible behavioral outcomes. The requirements to achieve optimal business impact frequently includes efforts toward a variety of inputs ranging from research, documentation, planning, refinement, testing, coding, reviews, releasing, monitoring, debugging, architectural design and many more. Software developers are expected to judge the appropriate level of effort to satisfy this complex and ambiguous set of requirements.

She puts TomTom's and the team's business impact first and finds ways to make this also her success. Being efficient and effective as a tech organization requires well-balanced³ collaboration, transparency by default, a bias towards early integration, reliable estimations and active management of unknowns, risks, constraints, and dependencies. She strives for "doing her part" and continuously improving her skillset in these critical, challenging, and diverse abilities. She understands the value of applying well-suited processes, policies, and governance on critical areas like cross-team alignment, cost efficiency, performance management, security, privacy, safety, and audit to ensure business health and effectiveness of the organization as a whole.

She takes pride in working at TomTom and takes ownership for and contributes to responsibilities beyond⁴ the scope of her team. In particular, she shows leadership solving customer and technical problems – leading the change – that leads to the most impact, and she is passionate about doing so, finding a way to make TomTom better for all of us.

Software Engineering Ladder

Level 13

Software Engineer 1

The Software Engineer 1 has graduate⁵ level programming expertise but little professional software engineering experience. She works on individual work packages under the guidance of more senior team members. Both she and the team invest in her growth – continuously working slightly outside of her comfort zone. Her primary growth goal is to deliver individual work packages and **small** product improvements (see appendix A: "About Product Improvements") with little guidance.

- Typically, "I" shaped skills⁶ with a dominant focus on one of the key software engineering disciplines (data engineering, server-side software development, native applications, web development, etc.)
- Showing accountable behavior for quality (and quantity) of own work (mainly code quality and fulfilling requirements), receiving support from more senior Software Engineers for successful delivery if needed

When to promote: Promotion should happen when the Software Engineer 1 is able to deliver small product improvements with little guidance and there are no behavioral or quality red flags.

³ Collaboration is expensive, and so we want to be deliberate in deciding where we want to collaborate, and where we do not. A platform team that is providing something as-a-service that requires collaboration is probably failing.

⁴ Depending on seniority of the role, please refer to individual grade descriptions for more details

⁵ This means she has a relevant bachelor's or master's degree or has gained comparable knowledge via other means. This knowledge includes, but not limited to programming language(s), principles, data structures, algorithms, databases, and data storage solutions. The litmus test for such knowledge is the ability to solve problems with code.

⁶ See "[Are You an "I" or a "T"?](#)"

Level 14

Software Engineer 2

The Software Engineer 2 has gained a solid software engineering skillset and has a foundational understanding of the product and the systems her team owns. She consistently delivers individual work packages and **small** product improvements with little to no guidance, knows how to operate the team's systems in typical situations and is an active member of the team's on-call rotation. She understands the impact of her own deliverables and actively contributes to solution and system designs within the scope of them.

She makes continual investments into her own growth and seeks to work on product improvements that push her outside of her comfort. She seeks out and is open to guidance from her teammates. She may require team support to continue to grow, but the time invested by her team does not exceed the value of her contributions.

- Limited T shaped skills,⁷ combining SW engineering fundamentals with deeper knowledge of specific technology
- Showing accountable behavior for successful delivery of own work
- Showing leadership by taking ownership of and improving small team-internal processes and ways of working or by leading small team-internal changes

When to promote: Promotion should happen when the Software Engineer 2 is able to deliver medium product improvements with little guidance, when she actively contributes to improving the operational aspects of the team's systems and to solutions and system design, and when there are no behavioral or quality red flags.

Level 15

Software Engineer 3

The Software Engineer 3 has gained a strong software engineering skillset and has a good understanding of the product and the systems her team owns. She consistently delivers **small to medium** product improvements with little to no guidance and contributes to improving the operational aspects of her team's systems. She estimates effort and identifies risks with reasonable accuracy. She actively participates in priority decisions and in solution and system designs beyond the scope of her own deliverables.

She takes ownership for her growth and seeks opportunities for working outside of her comfort zone. The team supports her on this, but she more than returns the time invested by the team through the contributions she makes to the team. She mentors more junior engineers selflessly and knows that success happens through the effectiveness of the whole team, not individual heroics. She contributes constructively to the community of Software Engineers primarily inside TomTom and occasionally outside.

- T shaped skills,⁸ where SW engineering fundamentals are coupled with deep knowledge of specific technologies
- Showing accountable behavior for successful delivery of own work
- Showing accountable behavior for fit-for-purpose implementation of processes, policies, and governance in the team

⁷ See "[Are You an 'I' or a 'T'?](#)"

⁸ See "[Are You an 'I' or a 'T'?](#)"

- Showing leadership by taking ownership of and improving medium team-internal processes and ways of working or by leading medium team-internal changes

When to promote: Promotion should happen when the Software Engineer 3 is able to solve all engineering problems in the scope of her team with no guidance, delivers significant impact to customers and the business, puts the success of the team above her own success and there are no behavioral or quality red flags.

Level 16

Software Engineer 4

The Software Engineer 4 can solve most or all the engineering problems in the scope of her team without guidance. She estimates effort and identifies risks with competence and confidence and drives technical and priority decisions that reinforce the product strategy and bring value sooner. She takes ownership of the team's success – supporting more junior team members in delivering on expectations, actively adjusting her own priorities as needed and leading change and innovation to continuously improve the team's ability to make an impact.

Because most of the engineering problems worth solving can be solved by Software Engineer 4, there is no hard limit⁹ to the impact she can deliver by continuing to learn and deepen her expertise, both on the domain and on her engineering skillset. There is no expectation that she strives for further promotion into either the management track or to Staff Software Engineer. She understands that she needs to own both her growth and finding opportunities to have impact if she deliberately decides to pursue that path.

- TT shaped skills, where SW engineering fundamentals are coupled with deeper knowledge of multiple, or connected technologies
- Accountable for successful delivery of own work, actively supporting other team members work and growth
- Showing accountable and influencing behavior for fit-for-purpose implementation of processes, policies, and governance in the team by being a true role model
- Showing leadership by taking ownership of major team-internal responsibilities, leading major team-internal changes, or small team-external responsibilities/changes
- Supporting hiring for team vacancies, and investments into team and product health

Software Engineering Ladder: Staff+¹⁰ Levels

The Staff+ Software Engineering ladder is not a direct continuation of the Software Engineering ladder. In fact, working as Staff+ Software Engineer can be considered a different job – just like Software Engineering Management is a different job – compared to the job as Software Engineer. A strong Software Engineering skillset is a pre-requisite for both however, so both roles pose an opportunity for Software Engineers who want to pivot their career towards leadership.

The Staff+ Software Engineer excels at leadership, impact, and capabilities. She consistently pushes the boundaries of what is possible, actively shapes the strategic direction of her team and beyond, and she inspires others to follow her. She identifies opportunities to enhance existing or create new businesses, significantly improve our systems landscape or ways of working – and she makes these

⁹ TBC: There is also no hard limit to the compensation she receives in line with the impact generated.

¹⁰ Staff+ refers to all Software Engineer levels 17 and above.

changes happen. She is a true corporate citizen, advocating for those standards, rules and governance that help the organization to scale. **She knows, it is not about her – instead, she takes pride in helping others around her, being successful and being a multiplier.** She actively works to avoid becoming a single point of failure.

Enabling cross-functional and cross-team groups to deliver highly complex projects is a core part of how she helps TomTom win. And while many of her actions on this might be invisible, the impact that she enables needs to be clearly visible¹¹. She shows accountable behavior while seeking opportunities to bring dramatic impact with tangible results.¹²

The expected level of impact can be delivered in a variety of different ways. To account for that fact, and to avoid “one size fits all” thinking in how we assess and grow potential, we define different archetypes of Staff+ Software Engineers. This list of archetypes should give an indication only, however. If a Staff+ Software Engineer can deliver the expected level of impact in another (software engineering) way, she will be able to thrive in our organization. Regardless of the archetype, most Staff+ Software Engineers will have a TTTT shaped skill set, with deep expertise on several -- often connected -- technologies across the whole stack. Staff+ Software Engineers rely on their coding skills and use them when and where applicable, balancing hands-on work with impact through others.

No matter how she delivers her primary impact, she takes ownership of building a world-class engineering organization together with other leaders throughout TomTom and actively contributes to our community of engineers.

The Role of Managers

Staff+ Software Engineers are true game changers and can give TomTom a competitive advantage like no other role in tech organizations. They “make great things happen” much faster than expected – or even make them possible in the first place. Identifying engineers that can be successful in such roles, supporting their growth and getting them on the most valuable challenges is fundamental to what a strong Engineering Management organization should strive for.

Archetypes

Tech Lead

The Tech Lead¹³ is the most common Staff+ archetype. The Tech Lead excels at ensuring clear goals are defined and everyone is working with sharp focus towards them. She easily handles frequent context switches, finds her way through “information overload” and drives collaboration with involved parties to design “fit for purpose” solutions and to break down work into manageable deliverables.

The Tech Lead is capable of driving complex programs of interrelated projects and is masterful at managing and mitigating the risks to ensure the successful and on-time delivery of our most critical software deliveries. She has a great network and always finds ways to unblock her team through

¹¹ Impact comes in many forms, making it “visible” can mean very different things. Where possible we expect the impact to be measurable. If that is not possible, it is the managers' accountability to be able to articulate why that particular delivery has the expected or assumed level of impact.

¹² As a consequence, promoting or hiring a Staff Engineer where significant impact is infeasible, impractical, or immaterial to the business is setting that Staff Engineer up to fail. The conditions must be strategic and at sufficient scale for a Staff Engineer role to find sufficient traction to justify it.

¹³ The term “Tech Lead” refers to the informal role of leading a certain scope/project/... to success, not to the often-used formal title. One, multiple or no tech leads can be on any given team.

creative pragmatism or conceptual breakthrough as the circumstances demand. She has a strong bias towards action, keeps stakeholders informed, simplifies the complex, values early integration, and pushes hard to get things across the finish line.

Architect

The Architect uses her deep understanding of her team's product and system architecture and dependencies to ensure and improve product health¹⁴, cost efficiency and the system's evolvability. She understands TomTom's strategy and shapes her team's product and technical strategy to align well with it – and applies her strategic abilities on supporting other Software Engineers in making decisions that not only solve today's problems, but also allow her team to adjust to tomorrow's problems.

She knows that the hallmark of good software design is a shared mental model everywhere it's needed and is skilled at generating appropriate mental models that everyone can understand and assimilate without ambiguity. She knows that "simple and working" is more flexible than "flexible and not working," and that the hallmark of good architecture is the ability for a system to adapt to unforeseen requirements, yet she avoids speculative design. She is skilled at working with teams to help them understand and implement what may feel like excess local complexity when doing so reduces the global complexity of a bigger system.

Her ability to identify opportunities for simplification and her focus on the non-happy paths of solutions are well respected by her team members and they seek her involvement in complex deliverables. She collaborates well and seeks to generate buy-in without ever shying away from constructive conflict.

Solver

The Solver takes ownership of the hardest problems that need to be solved – usually those problems that cannot be broken down into smaller sub-problems easily. These can be nasty and hard to reproduce bugs or very complex algorithmic or infrastructural challenges. She excels at analyzing these problems quickly, communicating complex matters in easy-to-understand ways and coordinating successful resolution.

With a strong focus on enabling her team to own the solution after successful delivery and on growing the analytical competencies of her peers she can act as a true multiplier. It is common for the Solver to rotate teams frequently to where she can have the most impact.

Researcher

The Researcher has strong analytical skills and is eager to explore new concepts and technologies. She can digest and explain complicated concepts to colleagues with different roles and backgrounds. A researcher excels at identifying and solving technical challenges that allow TomTom to remain a key player in the field. She is involved in developing proofs of concepts that help reduce engineering risk, develop patentable technology, and provide input for roadmaps. While most of her contribution happens in the exploratory phase, she understands that impact only comes through evolution of concepts into products.

She also has good technical writing skills.

¹⁴ Continuously meeting quality requirements of the product.

Levels

Level 17

Staff Software Engineer¹⁵ 1

The Staff Software Engineer 1 embraces her role as a multiplier, striving to up-level her team and everyone around her. She will typically enable a significant uplift on the impact generated by joining a common small engineering team. She usually does this by helping the team to deliver on the most valuable opportunities, identifying new opportunities for great impact and driving up the efficiency of the team – with very high variation how different Staff Software Engineers make it happen.

- Enabling delivery of product improvements that are beyond “Large,” emphasis on the “Impact” dimension
- Key contributor to the team's strategic direction, influence on the organization's direction
- Consistently identifying opportunities to improve the team and product and making it happen, not relying on others to “bring the problem” – define the what, the why and the how
- Respected as enabler for success and individual growth by team members
- Additionally taking ownership of initiatives and having impact outside of the team’s scope¹⁶

Level 18

Staff Software Engineer 2

The Staff Software Engineer 2 will typically enable an uplift on the impact generated by joining a small organization consisting of multiple teams. Guides and influences others either internally or externally to adopt a different point of view. Her work will often contribute significantly to the most important priorities of her unit.

- Enabling delivery of multiple product improvements that are beyond “Large,” emphasis on the “Impact” dimension – either directly or indirectly (e.g., by identifying and solving systemic technical or organizational challenges)
- Key contributor to the organization’s strategic direction, influence on the PU’s direction
- Ensures technical direction and decisions support company objectives and strategy – timely delivery of lasting results
- Respected as enabler for success and individual growth by peers throughout TomTom

Level 19

Principal Staff Software Engineer

The Staff Software Engineer 3 consistently achieves results that get awareness by top-level management. This can be significant uplifts on group-level metrics, realizing great cost savings, massive productivity, or governance improvements, identifying new opportunities, and successfully creating significant business out of them, or anything else with comparable impact. Persuades

¹⁵ This is following (and extending) the naming from <https://staffeng.com/guides/overview-overview> and <https://www.levels.fyi/?compare=Apple,Amazon,Google,Facebook,Microsoft&track=Software%20Engineer>.

¹⁶ Initiatives are usually unrelated to coding: mentoring, moderating OKR definition, contributing to a new PU strategy, etc.

leaders to make tough decisions and act upon them and/or negotiates with external partners/vendors/customers.

Level 20

Executive Staff Software Engineer

The Staff Software Engineer 4 is a competitive edge in the industry. Contributes to the company's technology direction and improves how the organization engineers and leverages technology.

Appendix

A. About Product Improvements

We use the term “product improvement” to refer to any sort of engineering work, whether it is new features, bug fixes, resolving technical debt or anything else that either improves the product directly, mitigates a risk to our product or improves our ability to deliver a great product in the future. Delivering a product improvement also always implies that this is delivered into production and available to/used by actual customers. Aspects like testing, monitoring, alignment with other teams, rollout plans and everything else required for successful delivery are always implicit.

The impact created through successfully delivering product improvements is the major dimension in assessing the impact (and level) of a Software Engineer. The table below aims at establishing a framework for assessing the size of a product improvement in line with the above descriptions for levels 13 through 16.

How to use the table: For every successfully delivered product improvement the Engineering Manager classifies the listed dimensions as Small, Medium, or Large (or in between, like Small/Medium). Out of these an overall classification is derived based on the EMs understanding. We go for applying judgement and openness to being challenged on this assessment, rather than for a hard average/formula/heuristic because we understand that the EM will have more context than any framework can derive.

Dimension	Small	Medium	Large
Impact	Visible uplift on team-level metrics, minor improvement to system health or minor deliverable on customer commitments	Significant uplift on team-level metrics, visible uplift on PU-level metrics, major improvement to system health or major deliverable on customer commitments	Significant uplift on PU-level metrics, critical improvement to system health or critical deliverable on customer commitments
Complexity	One component (service, application, ...) and few different technologies (programming languages, storage systems, ...)	Few components and a few different technologies	Many components, many different technologies
Collaboration	Team-internal collaboration only	Collaboration with cross-functional stakeholders from the same domain or members of other product teams	Collaboration with cross-functional stakeholders from other domains or members of many different product teams
Effort	About 2 engineer weeks	About 6 engineer weeks	About 2 engineer quarters
Ambiguity	Clear plan of implementation, no unknowns	Well-understood approach, some known unknowns	Analysis or experimentation required, many known unknowns, some unknown unknowns
Blast Radius ¹⁷	No customer impact, minor impact on operational cost and engineering effort	Minor customer impact, major impact on operational cost and engineering effort	Major customer impact

¹⁷ Likely (negative) impact if this deliverable causes an incident.

