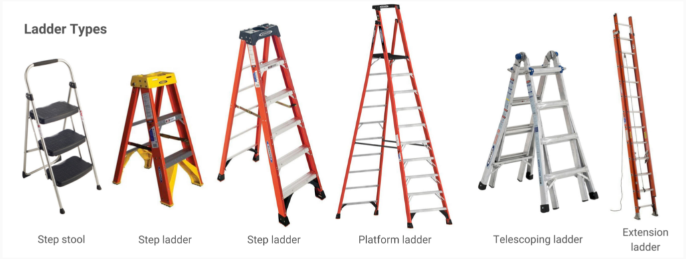
A Look at Software-Engineering Career Ladders

Petr Zemek

April 8, 2021

TIS (Threat Intelligence Systems) Seminar @ Avast

Ladder types (j/k)



https://safesitehq.com/osha-ladder-safety/

Outline

Before we begin...

Disclaimer:

- Everything in the talk is my own opinion
- I do not have all the answers
- Career ladders, job titles, and job descriptions are highly company-specific
- Any resemblance with existing ladders is purely coincidental ;-)

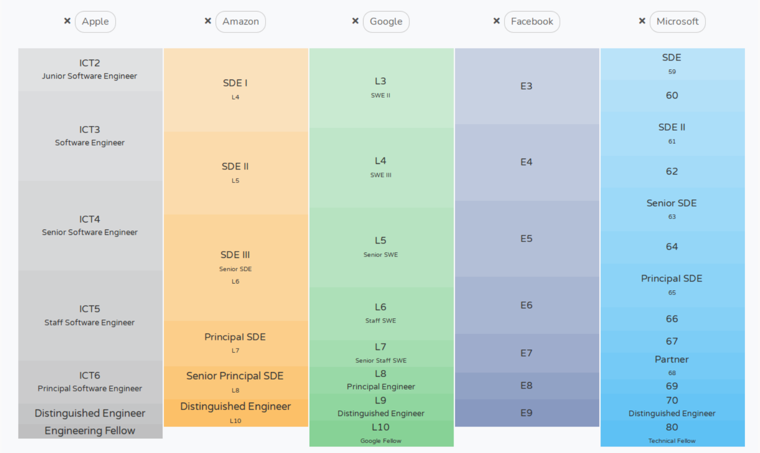
Programmers vs Software Developers vs Software Engineers

- Programmer
- Software Developer
- Software Engineer

https://softwareengineering.stackexchange.com/a/182147/96619

What is a career ladder and why is it needed?

What is a career ladder?



Apple	Amazon	Google	Facebook	Microsoft
ICT2 Junior Software Engineer	SDE I L4	L3 SWE II	E3	SDE 59
ICT3 Software Engineer	SDE II L5	L4 SWE III	E4	60
ICT4 Senior Software Engineer	SDE III Senior SDE L6	L5 Senior SWE	E5	SDE II 61
ICT5 Staff Software Engineer	Principal SDE L7	L6 Staff SWE	E6	62
ICT6 Principal Software Engineer	Senior Principal SDE L8	L7 Senior Staff SWE	E7	Senior SDE 63
Distinguished Engineer	Distinguished Engineer L10	L8 Principal Engineer	E8	64
Engineering Fellow		L9 Distinguished Engineer	E9	Principal SDE 65
		L10 Google Fellow		66
				67
				Partner 68
				69
				70 Distinguished Engineer
				80 Technical Fellow

https://www.levels.fyi/

Why do we need a career ladder?

A job ladder helps employees and well as the company.

- Engagement: People need to see that their careers can progress
- Provides clarity for individual contributors (ICs)
- Forces the company to become more clear in what it expects from people
- Provides a tool for managers (feedback, performance management, development)
- Consistent and more transparent promotion handling
- Helps with recruiting (hiring bar, apples-to-apples comparisons)
- Provides compensation ranges for HR

Initial remarks

- Identification
  - Numbers (e.g. L5 engineer)
  - Job titles (e.g. Senior Software Engineer)
- Senior *, I/II/III
- Internal titles vs external titles
- Single track vs dual track (IC/management) ladders
- Grade vs position vs role
- Generally, the higher up you go, the more ~~different~~ strategic your job is
- Promotion usually comes with a compensation adjustment
- Anti-pattern: Job-title inflation
- Anti-pattern: Job title not corresponding to the actual position
- Anti-pattern: Being promoted while doing the same work as before

Common questions

- How many levels there should be?
- Should compensation be directly tied to levels?
- Do level changes come with more responsibility?
- When should a person be promoted?
- Which levels are considered “final” (so-called *career positions*)?
- What does “senior” mean?
- Who is responsible for product and project management?

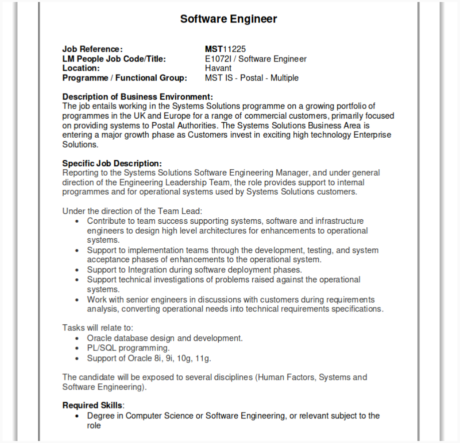
How can a ladder and position requirements be specified?

Three ways of specifying a ladder and position requirements

A diagram and:

1. Job descriptions
2. Snowflake model
3. Competency matrix

Job descriptions



Software Engineer

Job Reference:	**MST**11225
LM People Job Code/Title:	E1072I / Software Engineer
Location:	Havant
Programme / Functional Group:	MST IS - Postal - Multiple

Description of Business Environment:
The job entails working in the Systems Solutions programme on a growing portfolio of programmes in the UK and Europe for a range of commercial customers, primarily focused on providing systems to Postal Authorities. The Systems Solutions Business Area is entering a major growth phase as Customers invest in exciting high technology Enterprise Solutions.

Specific Job Description:
Reporting to the Systems Solutions Software Engineering Manager, and under general direction of the Engineering Leadership Team, the role provides support to internal programmes and for operational systems used by Systems Solutions customers.

Under the direction of the Team Lead:
- Contribute to team success supporting systems, software and infrastructure engineers to design high level architectures for enhancements to operational systems.
- Support to implementation teams through the development, testing, and system acceptance phases of enhancements to the operational system.
- Support to Integration during software deployment phases.
- Support technical investigations of problems raised against the operational systems.
- Work with senior engineers in discussions with customers during requirements analysis, converting operational needs into technical requirements specifications.

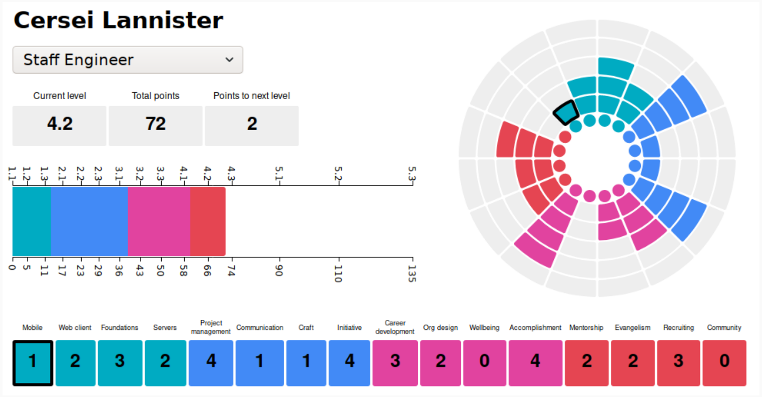
Tasks will relate to:
- Oracle database design and development.
- PL/SQL programming.
- Support of Oracle 8i, 9i, 10g, 11g.

The candidate will be exposed to several disciplines (Human Factors, Systems and Software Engineering).

Required Skills:
- Degree in Computer Science or Software Engineering, or relevant subject to the role

https://studylib.net/doc/6824639/software-engineer

Snowflake model



Cersei Lannister

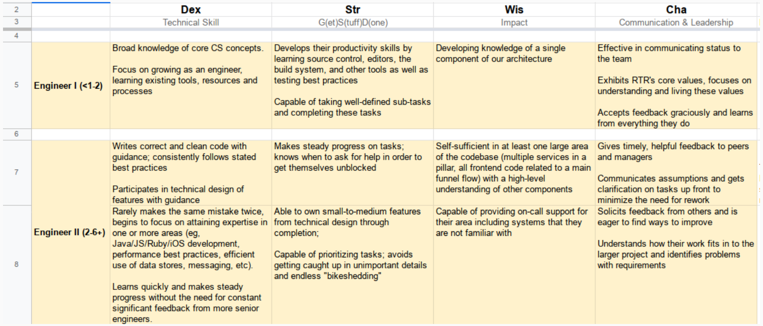
Staff Engineer

Current level	Total points	Points to next level
4.2	72	2

Mobile	Web client	Foundations	Servers	Project management	Communication	Craft	Initiative	Career development	Org design	Wellbeing	Accomplishment	Mentorship	Evangelism	Recruiting	Community
1	2	3	2	4	1	1	4	3	2	0	4	2	2	3	0

https://snowflake.medium.com/

Competency matrix

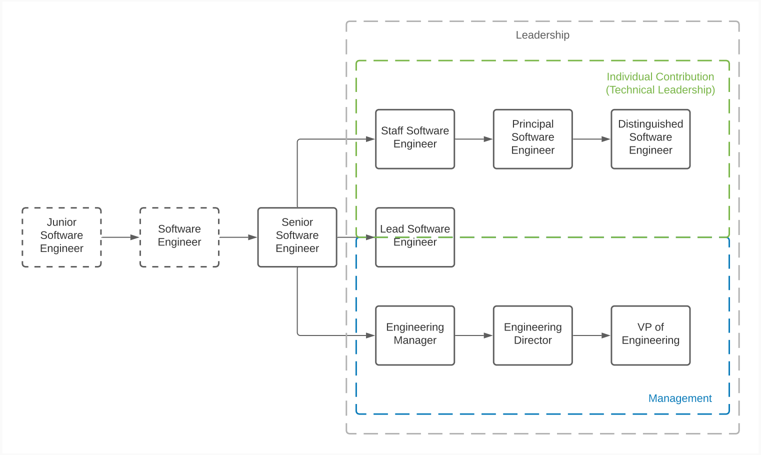


	Dex	Str	Wis	Cha
	Technical Skill	G(et)S(tuff)D(one)	Impact	Communication & Leadership
Engineer I (<1-2)	Broad knowledge of core CS concepts. Focus on growing as an engineer, learning existing tools, resources and processes	Develops their productivity skills by learning source control, editors, the build system, and other tools as well as testing best practices Capable of taking well-defined sub-tasks and completing these tasks	Developing knowledge of a single component of our architecture	Effective in communicating status to the team Exhibits RTR's core values, focuses on understanding and living these values Accepts feedback graciously and learns from everything they do
Engineer II (2-6+)	Writes correct and clean code with guidance; consistently follows stated best practices Participates in technical design of features with guidance	Makes steady progress on tasks; knows when to ask for help in order to get themselves unblocked	Self-sufficient in at least one large area of the codebase (multiple services in a pillar, all frontend code related to a main funnel flow) with a high-level understanding of other components	Gives timely, helpful feedback to peers and managers Communicates assumptions and gets clarification on tasks up front to minimize the need for rework
	Rarely makes the same mistake twice, begins to focus on attaining expertise in one or more areas (eg, Java/JS/Ruby/iOS development, performance best practices, efficient use of data stores, messaging, etc). Learns quickly and makes steady progress without the need for constant significant feedback from more senior engineers.	Able to own small-to-medium features from technical design through completion; Capable of prioritizing tasks; avoids getting caught up in unimportant details and endless "bikeshedding"	Capable of providing on-call support for their area including systems that they are not familiar with	Solicits feedback from others and is eager to find ways to improve Understands how their work fits in to the larger project and identifies problems with requirements

https://dresscode.renttherunway.com/blog/ladder

An example ladder with position overviews

An example dual-track career ladder

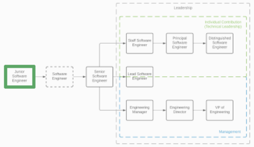


Notes

Usually, the higher you go:

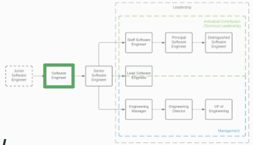
- The more power you have to influence things
- The more responsibility you have
- The more strategically you have to think
- The higher your compensation package is
- The more you have to deal with people
- The more meetings you have
- The less coding you do
- The fewer people on that position there are
- The more abstract the job descriptions are
- The more soft skills matter

Junior Software Engineer



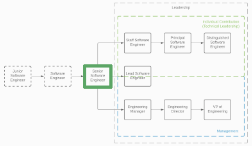
- Contribution: IC
- Years of experience: 0-2 (just for illustration)
- Gist: *An out-of-school person who has just started their career*
- Implements features, investigates and fixes bugs, writes tests, ...
- Requires lots of guidance to complete any non-trivial amount of work
- Main focus:
    - Learn engineering principles, practices, tools, and technologies
    - Learn how to communicate, ask for clarification, deliver feedback
    - Learn how to become more self-sufficient

(Mid-Level) Software Engineer



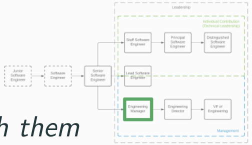
- Contribution: IC
- Years of experience: 1-7 (just for illustration)
- Gist: *Learning the ins and outs of the whole development life-cycle*
- Implements features, investigates and fixes bugs, writes tests, ...
- Can be given a well-defined ticket or small project, still requires guidance
- Main focus:
    - Learn the ins and outs of the whole development life-cycle
    - Improve development efficiency and quality of the produced work
    - Gain technical depth
    - Understand how their work fits into the larger picture
    - Learn how to become self-sufficient

Senior Software Engineer



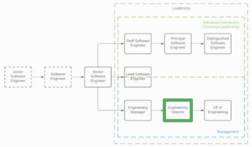
- Contribution: IC
- Years of experience: 5+ (just for illustration)
- Gist: *A seasoned, independent professional with deep technical knowledge*
- The workhorse of the engineering department, gets a lot done
- Can be given an epic or medium-sized project, almost completely independent
- Knows the used technologies inside out (depth orientation)
- Passive contributions outside of their own team (feedback, consultations)
- Mentors and trains new team members, shares knowledge

Engineering Manager



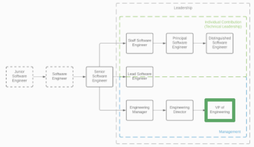
- Contribution: Manager
- Years of experience: 7+ (just for illustration)
- Gist: *Manages a team of 5-10 people (usually ICs), works through them*
- Cares for the well-being of the team, have regular 1:1s
- Writes performance reviews, manages performance issues, assigns bonuses
- Handles administrative tasks and communication with HR and IT
- Maintains budgets and head counts, handles salary increases
- Hires, promotes, and fires people
- Gathers product requirements, creates roadmaps, prioritizes
- Plans, assigns tasks to team members, specifies deadlines, monitors progress, evaluates results, provides feedback, eliminates roadblocks

Engineering Director



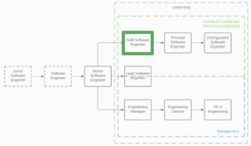
- Contribution: Manager
- Years of experience: 12+ (just for illustration)
- Gist: *Manages a team of managers*
- Similar responsibilities to Engineering Manager, but on a larger scale
- Responsible for multiple teams and many projects
- Has at least department-level impact

VP of Engineering



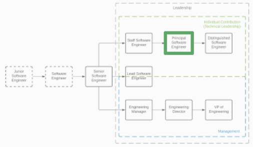
- Contribution: Manager
- Years of experience: 20+ (just for illustration)
- Gist: *Manages a team of directors*
- Responsible for everything that happens in the engineering department
- Company-level impact
- Pretty much on their own to figure things out
- Establishes the engineering culture and operations

Staff Software Engineer



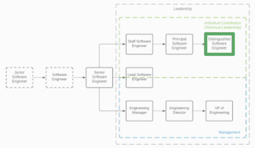
- Contribution: IC (technical leadership)
- Years of experience: 7+ (just for illustration)
- Gist: *A technical role model leading the design and development of larger projects*
- Does not have direct reports, leads by gravitas
- Has vast technological knowledge (breadth orientation)
- Has active cross-team contributions
- Leads the development of larger projects (Tech Lead)
- Leads the design of larger systems, components, and features (Architect)
- Solves complex technical issues (Solver)
- Writes specifications, reviews and consults technical proposals
- Mentors team members on higher-level technical issues (e.g. design techniques)

Principal Software Engineer



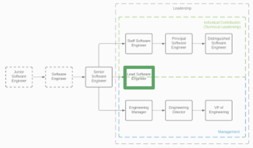
- Contribution: IC (technical leadership)
- Years of experience: 12+ (just for illustration)
- Gist: *A technical "director"*, *chief architect*
- Similar responsibilities to Staff Software Engineer, but on a larger scale
- Has extensive industry accomplishments or a key role in the company's technology
- Has at least department-level impact
- Identifies and works on company-wide strategic issues
- More high-level goals, e.g. "scale 100x"
- Sets technical directions, speaks for the company's technology

Distinguished Software Engineer



- Contribution: IC (technical leadership)
- Years of experience: 20+ (just for illustration)
- Gist: *A technical demigod*
- Industry-wide recognition (e.g. the creator of a programming language)
- Company-level impact
- Responsible for the technical strategy of all the engineering

Lead Software Engineer



- Contribution: IC/Manager
- Years of experience: 7+ (just for illustration)
- Gist: *Manages a team of 2-4 people while acting as an effective team member*
- A hybrid between a Staff Software Engineer and Engineering Manager
- Usually focuses more on the technical side than on the people side

Other positions and roles

- Project manager
- Product manager
- Scrum master, Product owner
- Architect
- Tech lead
- Team lead
- Senior * (e.g. Senior Engineering Manager)
- Fellow
- C-level executive (e.g. CTO)

Further reading and watching

Further reading (books)

- W. Larson: Staff Engineer: Leadership Beyond the Management Track (2021)
- D. Heller: Building a Career in Software (2020)
- R. Martin: The Clean Coder (2011)
- G. Weinberg: Becoming a Technical Leader (1986)

- W. Larson: An Elegant Puzzle: Systems of Engineering Management (2019)
- C. Fournier: The Manager's Path (2017)
- M. Lopp: Managing Humans (2016)
- P. Drucker: The Effective Executive (2006)

Further reading (articles)

- C. Groom: The software engineering job ladder
- P. Kua: The definition of a Tech Lead
- K. S. Tay: What is the etymology of the title Staff Engineer?
- P. Zemek: Not all developers want to be managers, and that's OK

- S. Joseph: Programmer competency matrix
- progression.fyi – A collection of public progression frameworks and career ladders
- levels.fyi – Salaries & tools to level up your career

Further watching

- M. Rogers: Creating a career ladder for engineers
- R. Koutnik: Rethinking the developer career path
- C. Engel: The software engineer career ladder explained
- C. Mihailescu: Career paths for software engineers
- C. Mihailescu: What is a Staff Software Engineer at Google?
- E. O'Neil: Congrats! You're the Tech Lead - now what?
- P. Kua: What I wish I knew as a first time Tech Lead