Software Teams: Roles and Collaborations

17-313 Fall 2023

Foundations of Software Engineering

https://cmu-313.github.io

Andrew Begel and Rohan Padhye

Administrivia

- We are grading your midterms throughout the week.
- Turn in your teamwork assessment surveys by Friday at 3:30pm.
- Enjoy Fall Break!

Learning Goals

- Software teams are comprised of more than just software developers. Those other roles are vital to creating, shipping, and selling software.
- Organizational hierarchies enable people of many different roles to collaborate together.
- Every role benefits from different personal strengths.
- Getting everyone to work together effectively is not easy.

Software Teams

Are Made of People

Team Roles Boggle

Working by yourselves, without talking, write down 10 *different* roles that people play on software teams. You have 5 minutes.

- I’ll give you one to start: Software Developer

Team Roles Boggle Group Discussion

- Now, pair up with a neighbor and compare your roles.
  You have 10 minutes.
  - Did you have any the others didn’t?
  - □n your group, discuss what activities people in each of those roles do, and what aspects of the software they are responsible for.

Team Roles Boggle Class Discussion

Report back to the class the roles you discussed and tell us the responsibilities of each role.

- The group with the most unique roles wins a prize!

Software Engineer

- Write the code
- Read and write development specifications
- Read, write, search technical documentation
- Fix bugs and debug code
- Write and run tests
- Check in and check out code
- Discover, find, install, run, and build tools
- Talk to other engineers and non-engineers

Architect

- Design the software architecture
- Enforce modularity across teams
- Audit AP□designs
- Write style guidelines
- Decide on internal tools

Program Manager

- Read and write design specifications / user stories
- Solicit and interpret the customer's needs and feedback
- Coordinate activities of all engineers
- Sets the timeline for implementing, testing, and shipping user stories

Tester

- Read and write test specifications
- Run tests (e.g., unit, module, integration, systems, acceptance)
- Develop and maintain testing infrastructure
- Push back on developers who write untestable code

Data Scientist (Newest Role)

- Collect
  - Build the data collection platform
  - □nject telemetry
  - Buildi the experimentation platform
- Use and Disseminate
  - Operationalize models
  - Define actions and triggers
  - Apply insights/models to business

- Analyze
  - Merge and clean data
  - Sample data
  - Shape data
  - Select features
  - Define sensible metrics
  - Build predictive models
  - Define ground truth
  - Test hypotheses

Working Styles of Data Scientists

Insight Provider

Specialists

Platform Builder

Polymath

Team Leader

Data Scientist – □nsight Providers

- Coordinate between managers and engineers
- Generate insights to guide managers in decision-making
- Strong communication and coordination skills

Data Scientist – Modeling Specialists

- Expert consultants
- Build predictive models to be product features
- Support other team’s decision-making
- Strong background in AI and machine learning

Data Scientist – Platform Builders

- Build reusable data engineering platforms
- Strong background in big data systems
- Makes tradeoffs between engineering and science concerns

Data Scientist – Polymaths

- Data scientists who “do it all”
- Form a business goal
- Instrument a system to collect data
- Do necessary analyses or experiments
- Communicate the results to managers

Data Scientist – Team Leaders

- Senior data scientists who run their own data science teams
- Act as data science “evangelists,” pushing for the adoption of data-driven decision making
- Work with senior company leaders to inform broad business decisions

User Experience Designer/Researcher

- Designs the user experience and the U□
- Runs user studies to identify positive and negative attributes of user experience
- Designs infinite variations of graphics and U□s for logos, buttons, sliders, text, layout, etc. to help engineering team decide on the look and feel of the product.
- Prototypes novel user experiences for future products

Operations Engineer

- Audit software deployments
- Maintain deployed systems
- Maintain server farms
- Monitor outages
- First to file incident reports

Release Manager

- Audits the software process for repeatability
- Verifies and audits security, privacy, and ethics properties of software
- Ensures software conforms with government regulations in every country product is deployed or shipped to

Sales Manager

- Finds potential customers for the product
- □dentifies customer wishlist features for product line
- Makes customers feel good about their purchases
- Establishes Service Level Agreements with customers (i.e., service contracts)
- Tries very hard to be ethical, e.g. don’t bribe the customer to buy the product, don’t wine and dine the customer at strip clubs, don’t make deals that stop customer from buying competitor’s products, etc.

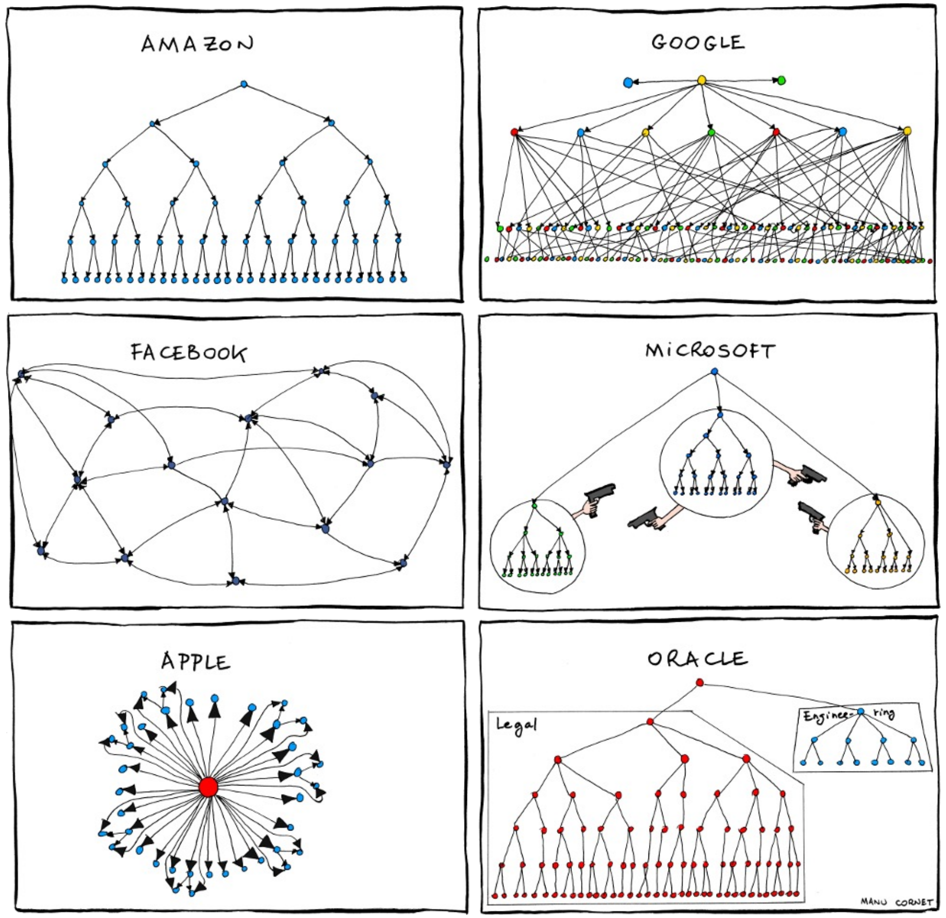
Legal

- Ensure all teams are following local, state, national, international laws and regulations
- Write intellectual property (IP) patents for offensive and defensive use
- Defend the company from lawsuits related to product use, workplace discrimination, IP theft
- Provide legal advice to product and sales team hypotheticals
- Provide an ethical backstop to inhibit questionable deals

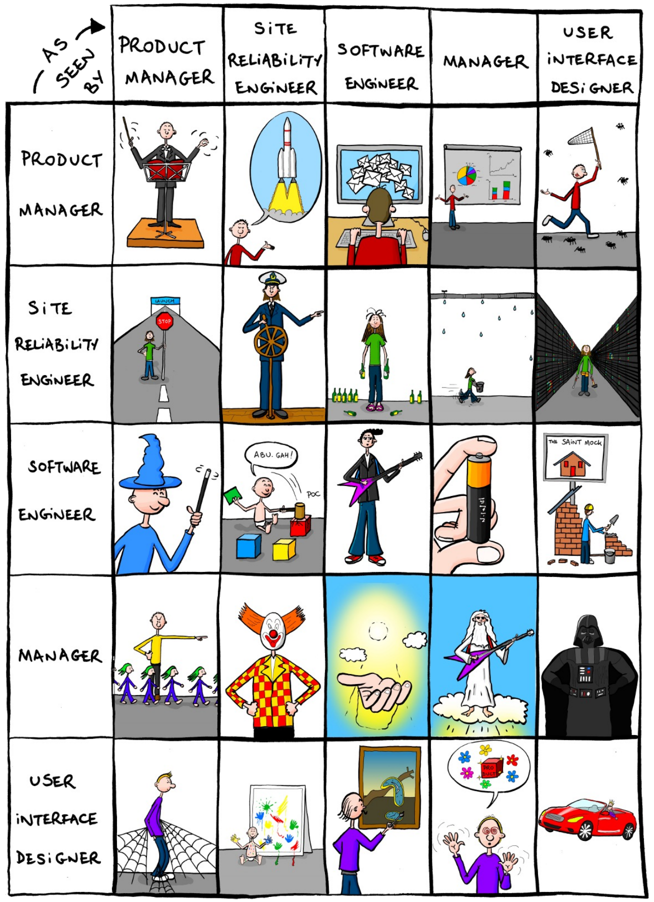
Product Manager

- Organizes all the employees who work together to design, develop, test, and ship the product
- Decides on the product’s North Star
- Decides the major feature pillars for each release
- Monitors development process and product telemetry
- Organizes (and reorganizes) staff into teams
- Hires and fires engineering managers, architect, product sales and marketing directors
- Responsible for profit and loss

Software Team Organizational Structures



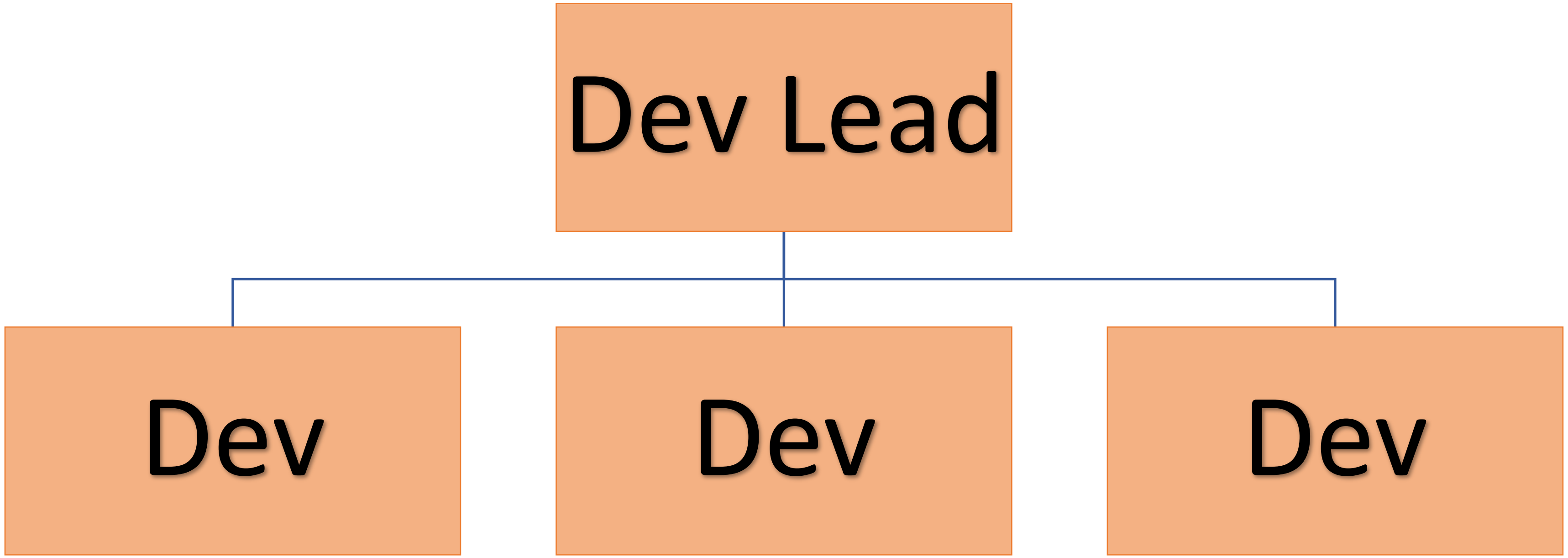
Can’t We All Get Along?



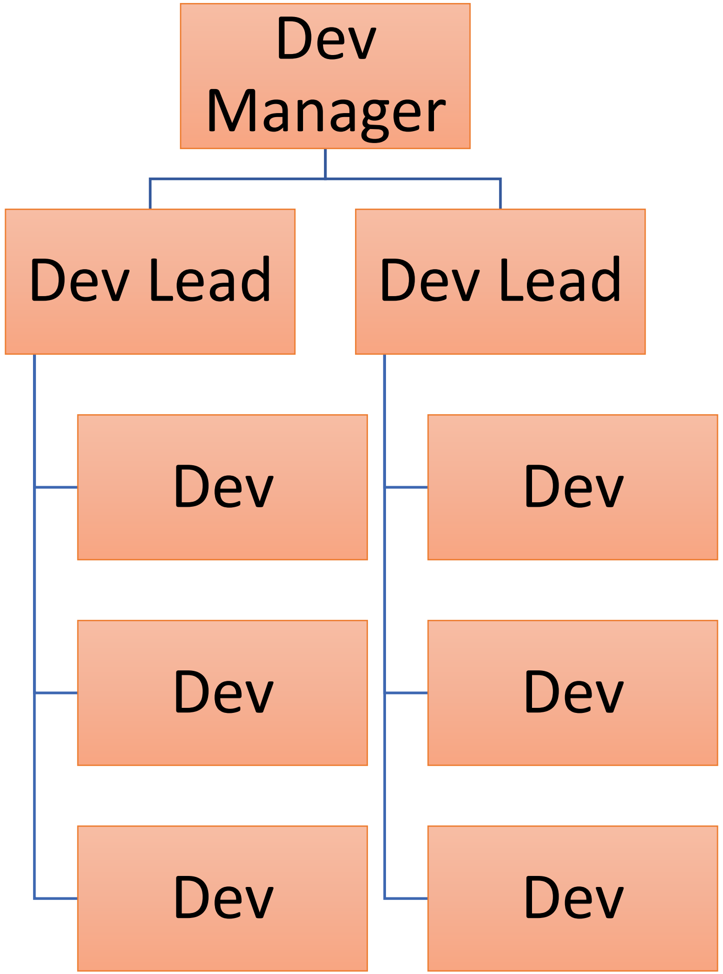
Organizational Structures

- Professional developers work closely with other developers, program managers, testers, and managers.
- They organize themselves into multiple simultaneous hierarchies.

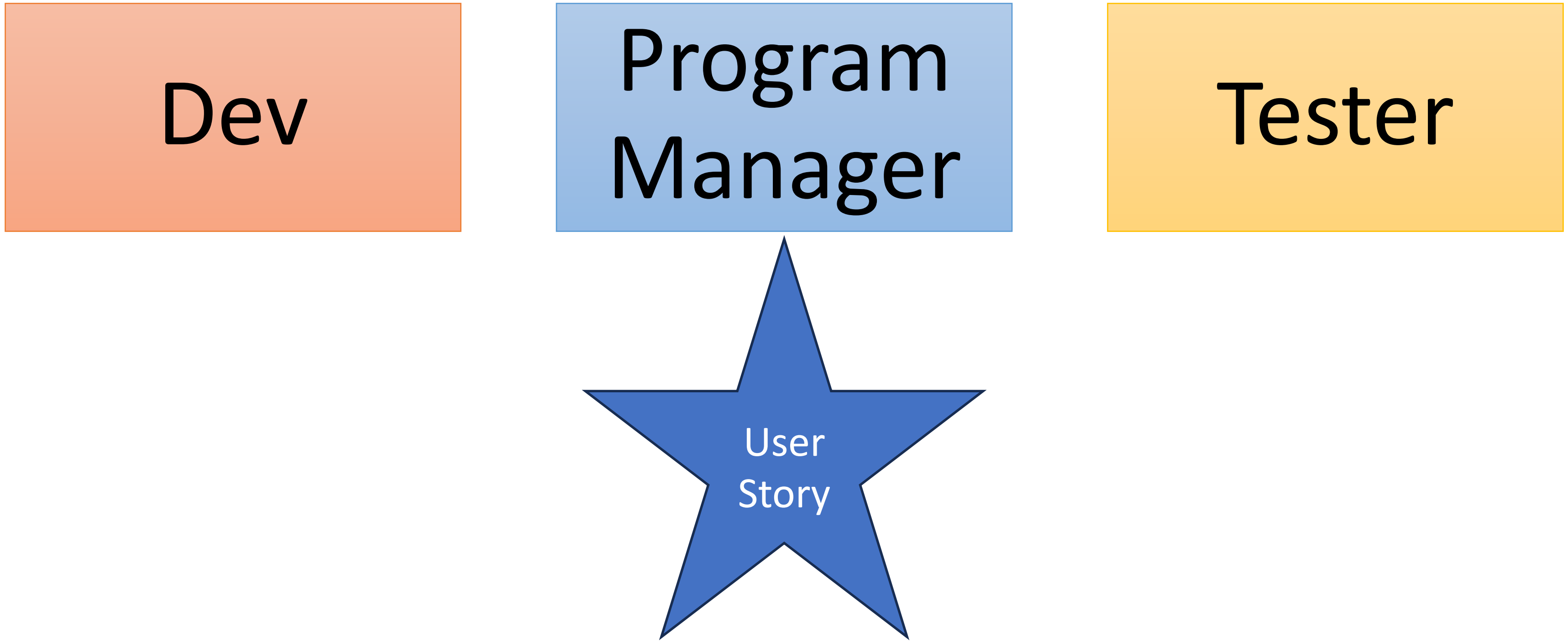
Microsoft Developer Hierarchies



Microsoft Developer Hierarchies



Microsoft Engineering Triad



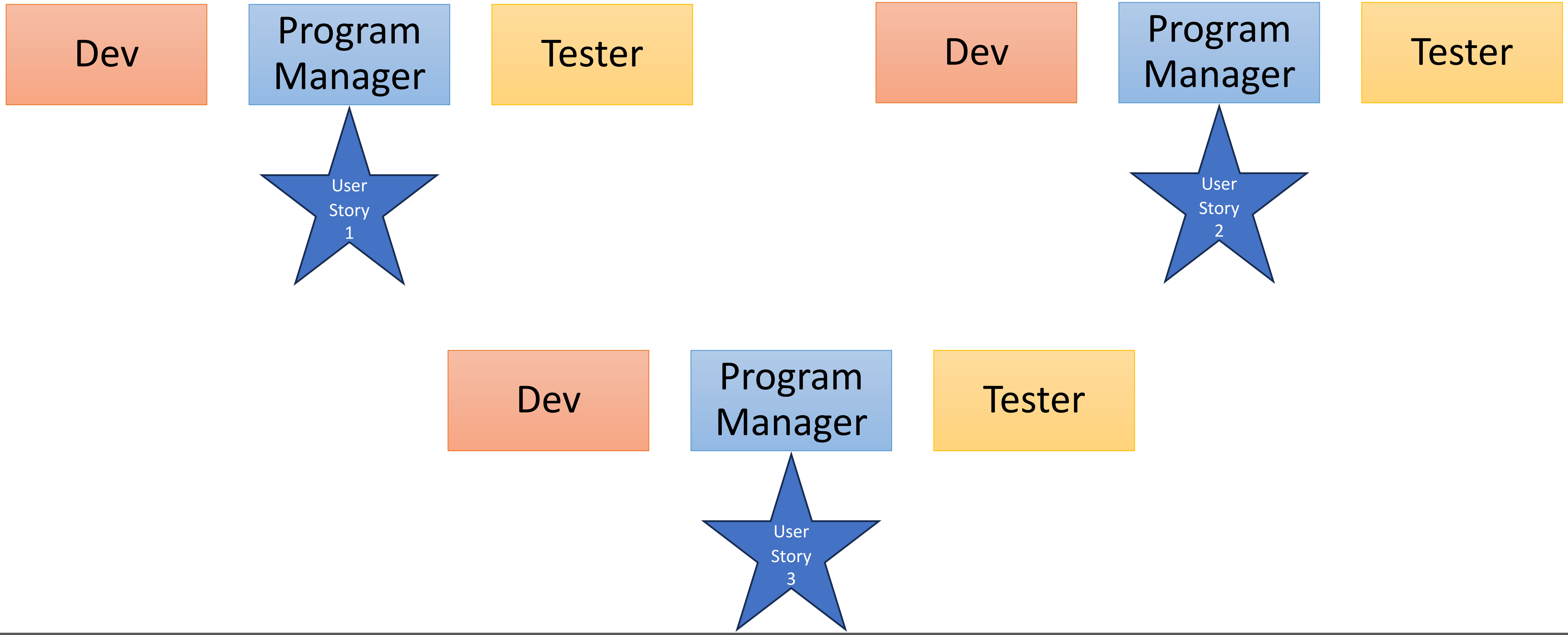
Dev

Program Manager

Tester

User Story

Microsoft Engineering Triads



Dev

Program Manager

Tester

User Story 1

Dev

Program Manager

Tester

User Story 2

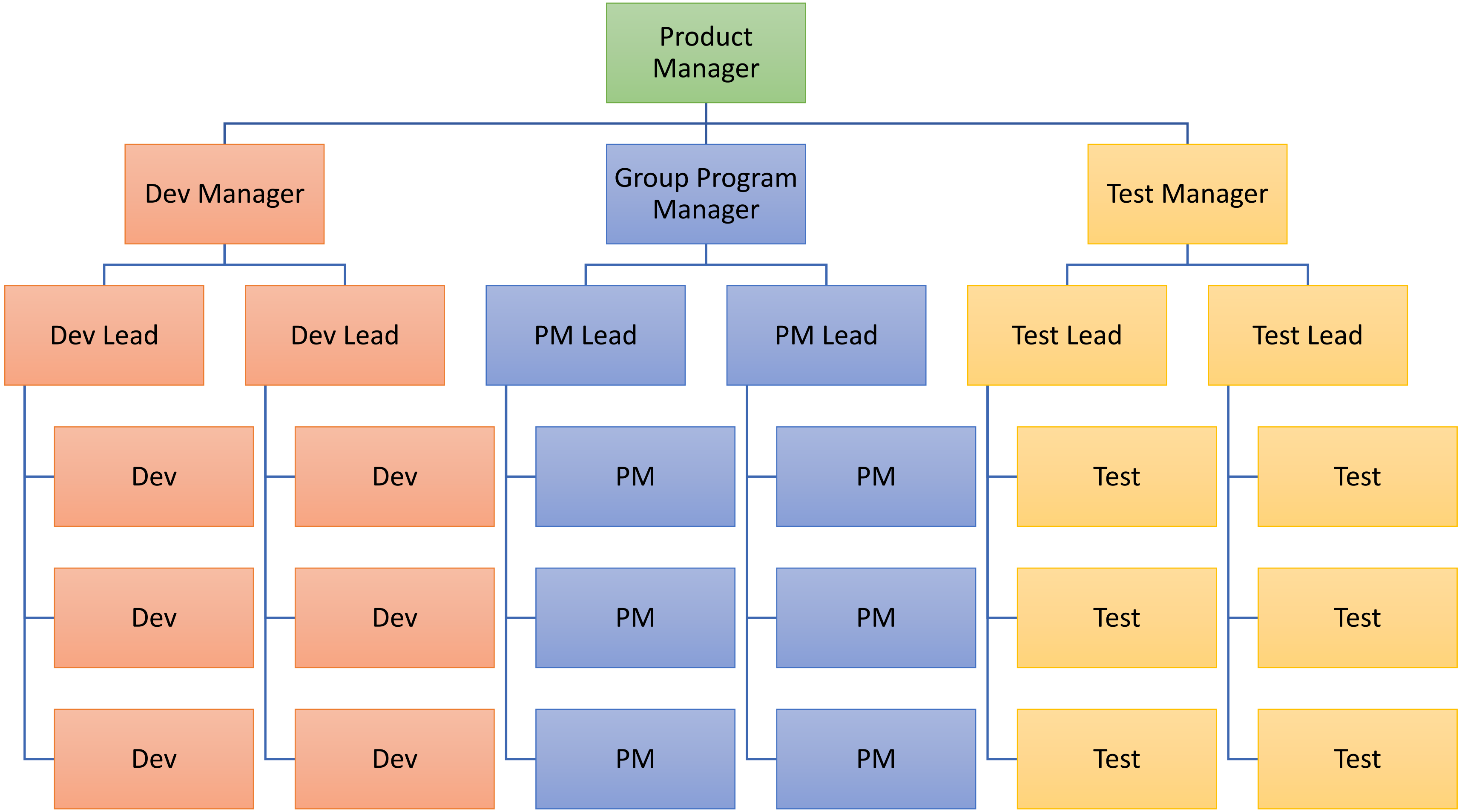
Dev

Program Manager

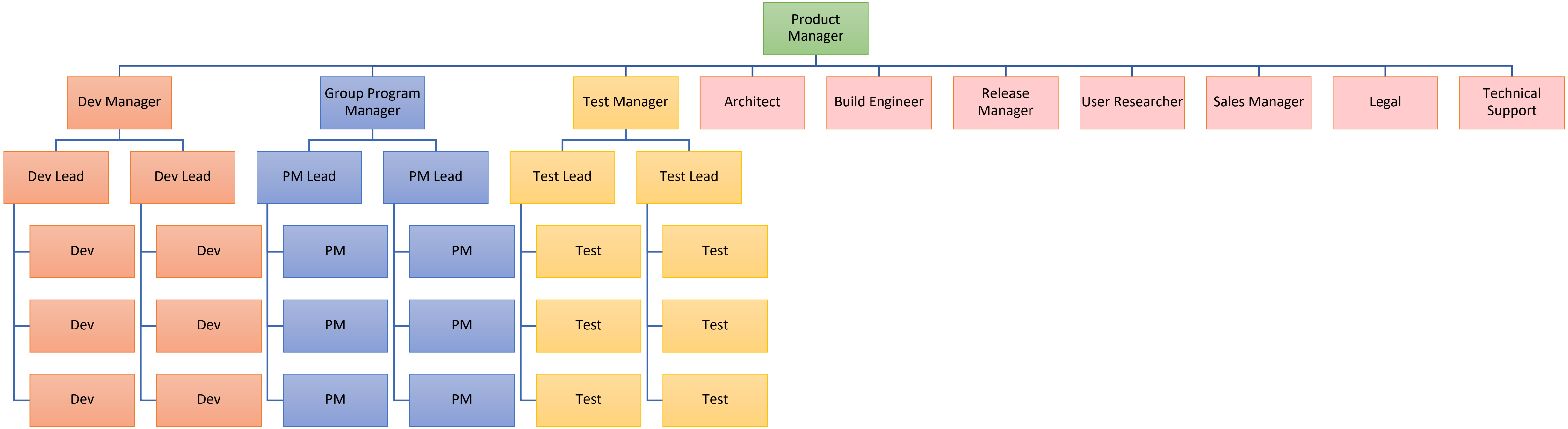
Tester

User Story 3

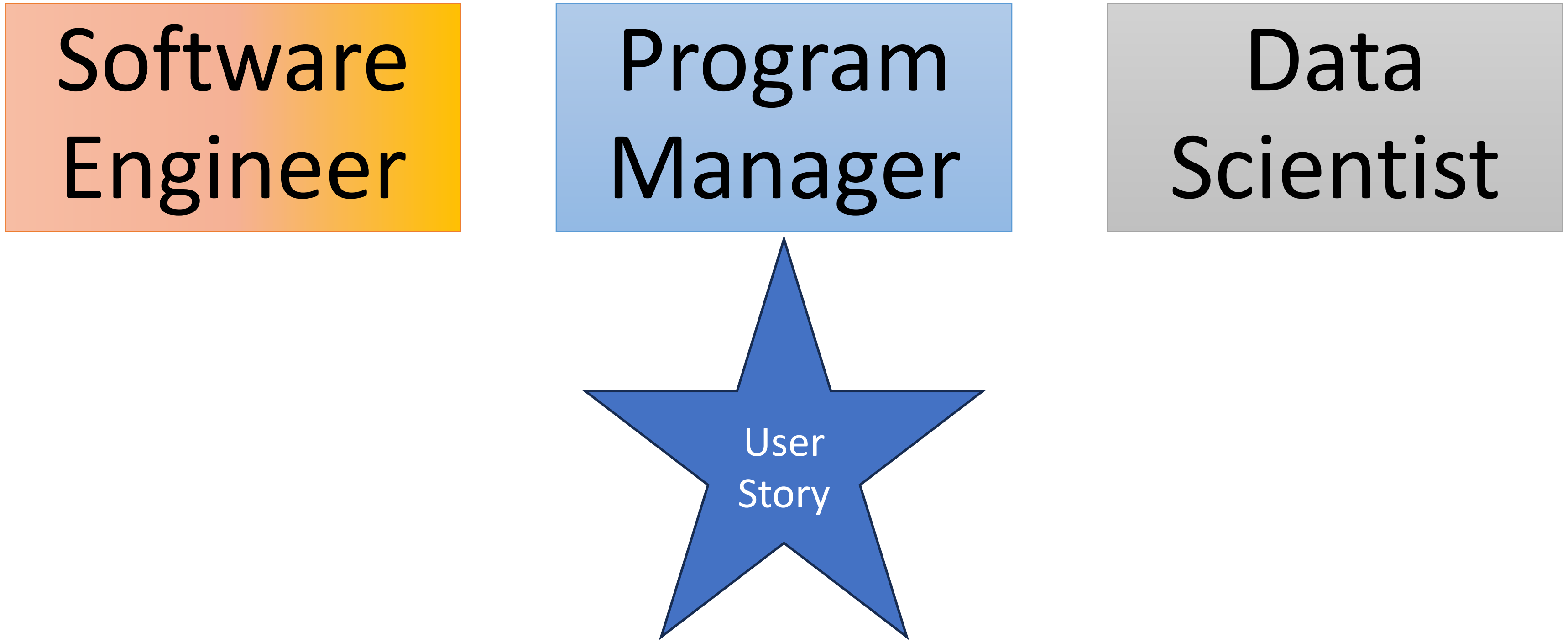
Microsoft Engineering Triad Hierarchy



Product Team Hierarchy



Microsoft ”New” Engineering Triad



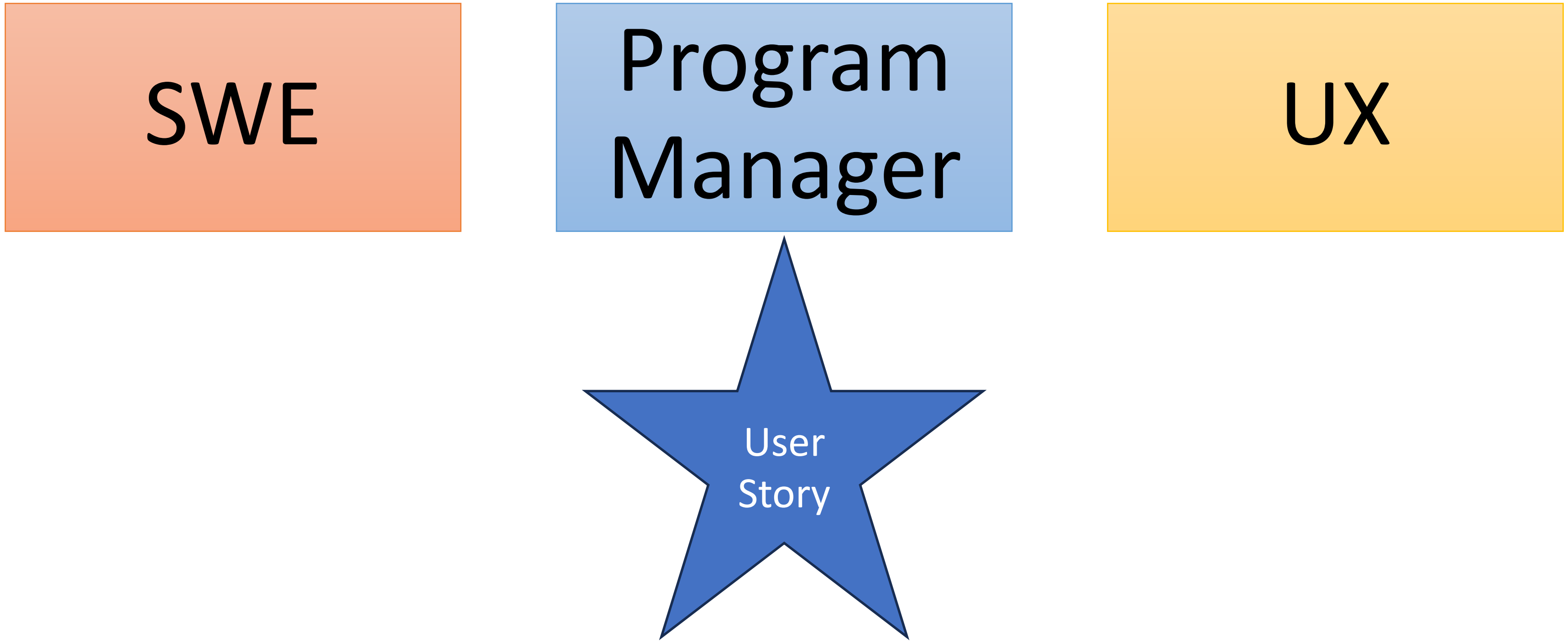
Software Engineer

Program Manager

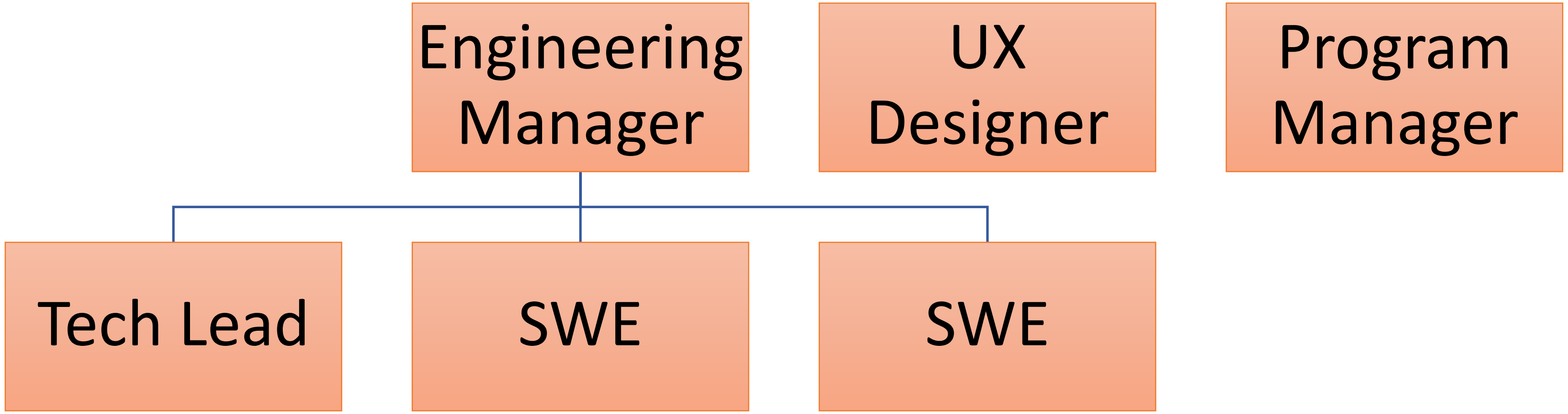
Data Scientist

User Story

Google Engineering Triad



Google Engineering Team



Engineering Manager

UX Designer

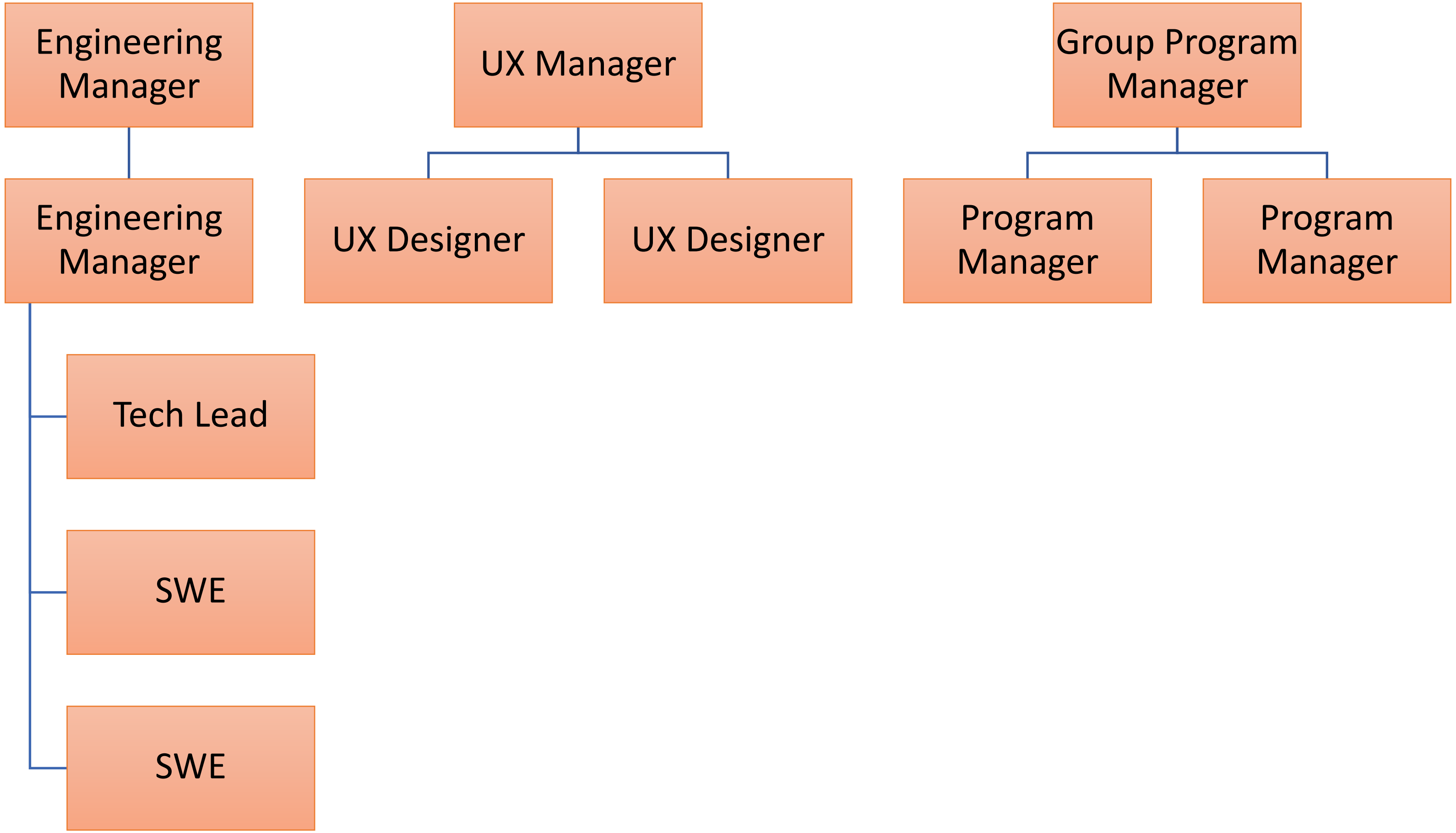
Program Manager

Tech Lead

SWE

SWE

Google Engineering Team Hierarchy



Engineering Manager

Engineering Manager

Tech Lead

SWE

SWE

UX Manager

UX Designer

UX Designer

Group Program Manager

Program Manager

Program Manager

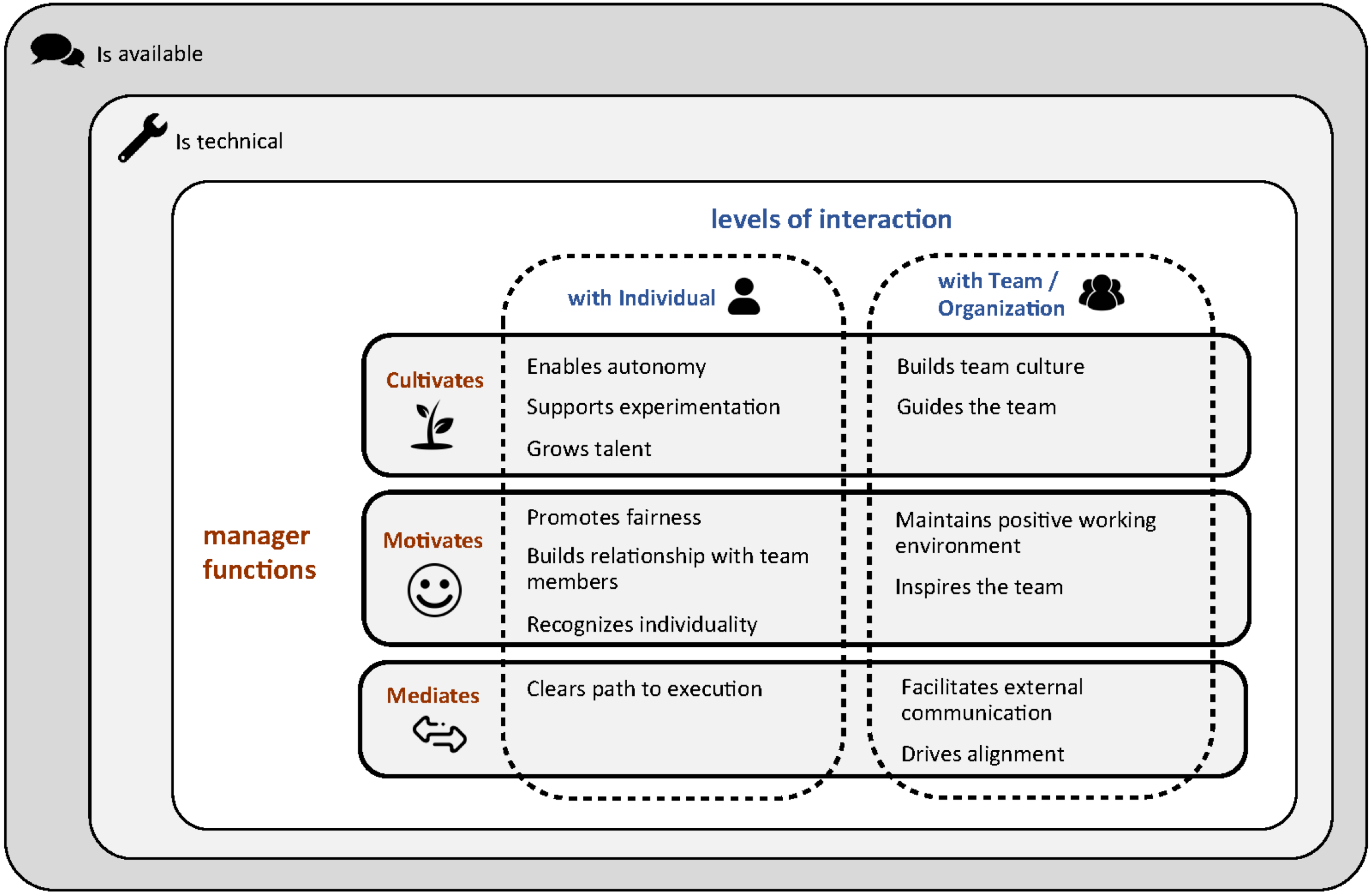
What Makes a Great Developer?

(as seen by developers)

1. Pays attention to coding details, i.e. methodical
2. Mentally capable of handling complexity
3. Continuously improves
4. Honest, i.e. provides credible information
5. Open-minded
6. Executes, i.e. gets things done
7. Self-reliant, i.e. can get things done independently
8. Self-reflecting, i.e. pivots when the plan goes awry

What Makes a Great Manager?

(as seen by developers)



Is available

Is technical

levels of interaction

	with Individual	with Team / Organization
Cultivates	Enables autonomy Supports experimentation Grows talent	Builds team culture Guides the team
Motivates	Promotes fairness Builds relationship with team members Recognizes individuality	Maintains positive working environment Inspires the team
Mediates	Clears path to execution	Facilitates external communication Drives alignment

What Makes Developers Good Collaborators?

(as seen by non-developer collaborators)

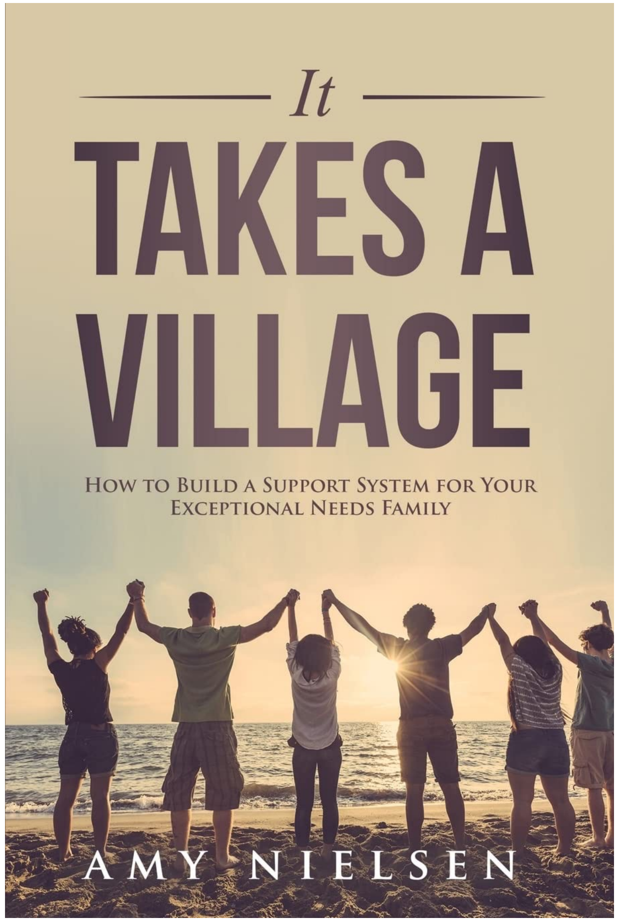
- Be expert in the code
- Be proactive communicators
- Developers don’t need to know everything. Every role makes a valuable contribution.

What Makes Developers Bad Collaborators?

(as seen by non-developer collaborators)

- Disrespect non-developer expertise
- Lack a can-do attitude
- Can’t handle multiple domains (e.g., software + hardware)

It takes teamwork to make the dream work!


It
TAKES A
VILLAGE
How to Build a Support System for Your Exceptional Needs Family
AMY NIELSEN