Disclaimer

I cannot speak for every workplace

I cannot speak for every team

I cannot speak for every manager

Team Roles

Team structures vary.

Some common roles:

- PM (sometimes)
- Engineering Manager
- Team Lead
- Developers

Roles often found outside your team

- Designer (usually)
- Quality Assurance/Test Engineer (technically)
- Database Administrator
- Ops
- Security

Why Team?

Your team is the most common center of work

- Sets priorities (!)
- Basic resources
- Feedback sources
- Context
- Common success/failure

You may have little idea of what other teams are doing

- the further from your team, the less you know

Team Roles

How do you interact with these people?

PMs

Product Manager (PM) or Project Manager (PM)

- Products are specific apps/tools/functionality
    - focus on making improvements, goals
- Projects are specific timelines and requirements
    - focus on goals with time/resource limits

PMs **may** know code, but often don't

- PMs **are** familiar with the topic and user needs
- Do not assume non-coding means ignorant

Interacting with PM

PM

- Sets definition of "done"?
- Sets task priorities
    - You should set task estimates
- Is often "client"
    - or represents client

You should not agree to unsustainable commitments

- They may need you to say what **is** sustainable

PM should be partnership

- Remember PM likely isn't your manager

Manager

Your direct boss (usually)

- Overrides PM
- Manages *team*
    - Includes interactions
- Training varies!
    - People skills are different skillset
- Often overbooked with meetings
    - Can get out of touch

Interacting with Manager

A good manager **removes** obstacles to your work

- Help them do that with information

1:1 (One-on-ones)

- A regular meeting with an authority figure
- Each of you can express hopes and concerns
- Should NOT be an evaluation
    - Do not feel you need to justify or defend
    - The idea is to meet regularly

Team Lead

This may the PM or Manager

- or may be a separate duty for a dev

Goal is to find "blockers" and resolve them

- keeps timeline accurate
- Not about people management
- But people skills valuable

Interacting with Team Lead

May soon be you!

Common Advice:

- Find what they are trying to do
    - Not what you want someone to do
- Help them do it
- Request when you need them to do it for you
    - Domain knowledge

Developers

Different types and levels and focus

- Mobile
- Frontend
    - UI
    - Middleware
- Backend

Often no real idea/training/guidance how to mentor

Different histories and personalities

Worry less about judgment and more about inclusivity

Dev Interactions

Not a competition, team event

Can learn from any dev

- Learn their specialties

Can teach any dev

- None of us know it all
- No one likes someone who discounts our experience

Don't assume they want their team role/focus

Non-Team Roles

These are roles that often interact with the team

- But are not (or often not) ON the team
- This changes the interactions
- Often have different priorities

Designer

Occasionally on the team

- Usually adjunct
- Have multiple teams

Designers are frequently:

- Key to a good product
- An easy source of conflict

May or may not have practical web UX experience

- Do they focus on how it looks or how it works?
- Your role is to work with, not obey or dictate

Quality Assurance (QA) aka Test Engineers

Vital source of domain information

- NOT there to find your bugs (you should!)
- They make you look good
    - or reveal you are bad

Often "assigned" to team

- but don't report to manager
- Not quite same priorities

Juggle an astounding number of languages/frameworks/services

Database Administrator (DBA)

- Likely exist outside your team
  - May not exist (!)
- Often their priorities aren't yours

Databases are vital

- data
- speed
- reliability

DBAs work out problems

Operations

"Ops" / "Admins"

Almost never part of a dev team

Maintain the network, operating machines, and fundamental software installations

- They DO NOT have your priorities
- Their jobs are important

Security

Almost never part of a dev team

- They do not share your priorities
- They tend to lean harder on the uncooperative
- Like QA, do your part before they are involved
- Very easy to feel like they are only a problem
    - Fight this urge