PRODUCT MANAGEMENT

PDM 5.0

Working with Developers

GENERAL ASSEMBLY

Working with Developers

LESSON ROADMAP

Welcom + Warm-Up

Navigating the Tech Stack

Communicating with Engineers

Managing Last Minute Development Requests

Bring It Home

LEARNING OBJECTIVES

1. Discuss how to ask technical questions and develop confidence in doing so.
2. Describe best practices for working with cross functional teams.
3. Explain ways to adjust a development plan.

NAVIGATING THE TECH STACK

Working with Developers

Welcome+ Warm-Up

Navigating the Tech Stack

Communicating with Engineers

Managing Last Minute Development Requests

Bring It Home

SOLO EXERCISE: TECH TERMS RECAP

10 Minutes

Whew! We covered a lot of new technical terminology in this section.

On your own:

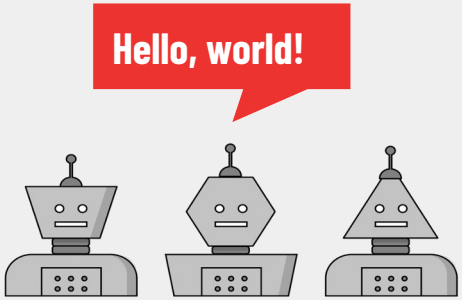
In Slack, write down one term or concept on which you need more clarity. Look at what others have written. **Can you explain what any of them are about? Do you need clarity on a term, too?**

Add a reaction: ➕ Need clarity ✅ I could explain it!

As a class:

We'll discuss and define some of the terms that are still unknown.

You don't *need* to be an Engineer...



...but as a PM, it's your job to:

- Support engineers in diagnosing and fixing technical issues.
- Share the information your engineers need in order to build great features.
- Craft clear messages for stakeholders and users.

Stuff Developers say

Heads up, we seem to be getting 503s on some yarn packages on admin. This is failing deploys to dev apps and staging.

It would be cool if we could just have a singular assessment component where we pass in props like `onQuestionSubmit: ()=>{} ``currentPage: 'score-report'`, regardless of URL structure.

Amazon is investigating reports of occasional DNS resolution errors with Route 53 and the external DNS providers. Amazon is actively working toward resolution.

Looking to do some benchmarking on create-api to test out our course-versions endpoints. Is it OK if we sync the sandbox db with the prod db?

Dev speak, translated:

The website is down so we can't fix stuff.

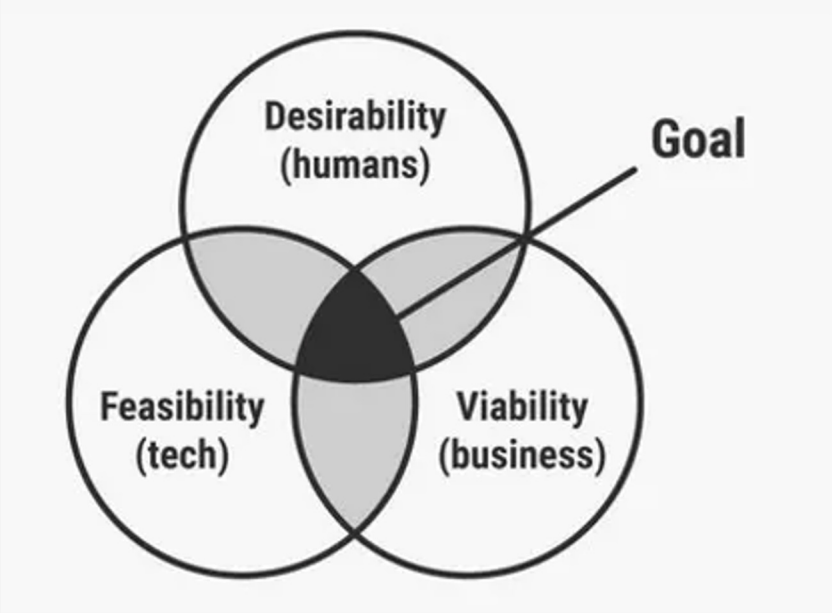
Here's a functionality that could make the Assessments product better.

The website is down.

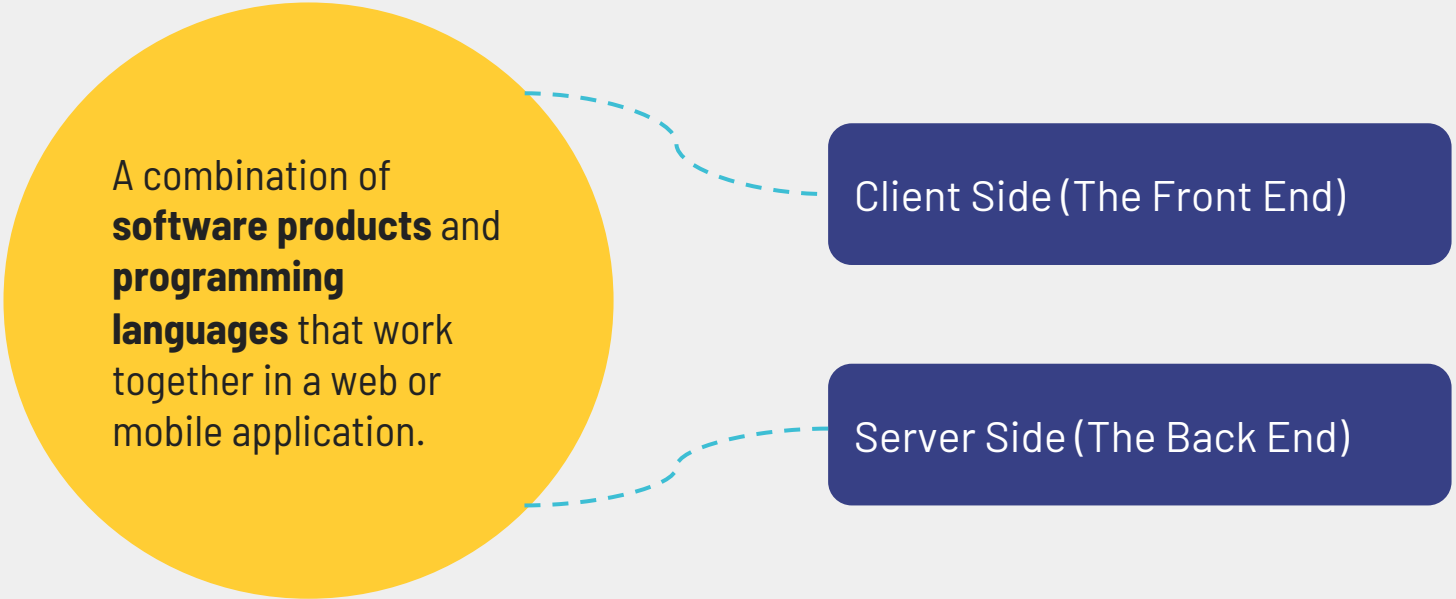
I want to test out a new functionality for the Create product.

Collaborative Trio (again!)

- ➔ Security
- ➔ Scalability
- ➔ Reliability
- ➔ Efficiency
- ➔ Productivity



THE TECH STACK AKA “THE STACK”



The Front-End/Client Side

- ➔ Powers the visuals and interactions the user sees
- ➔ The structure and style of the website or application
- ➔ Includes accessibility, performance, cross-browser, and cross-device functionality

HTML, CSS, JavaScript

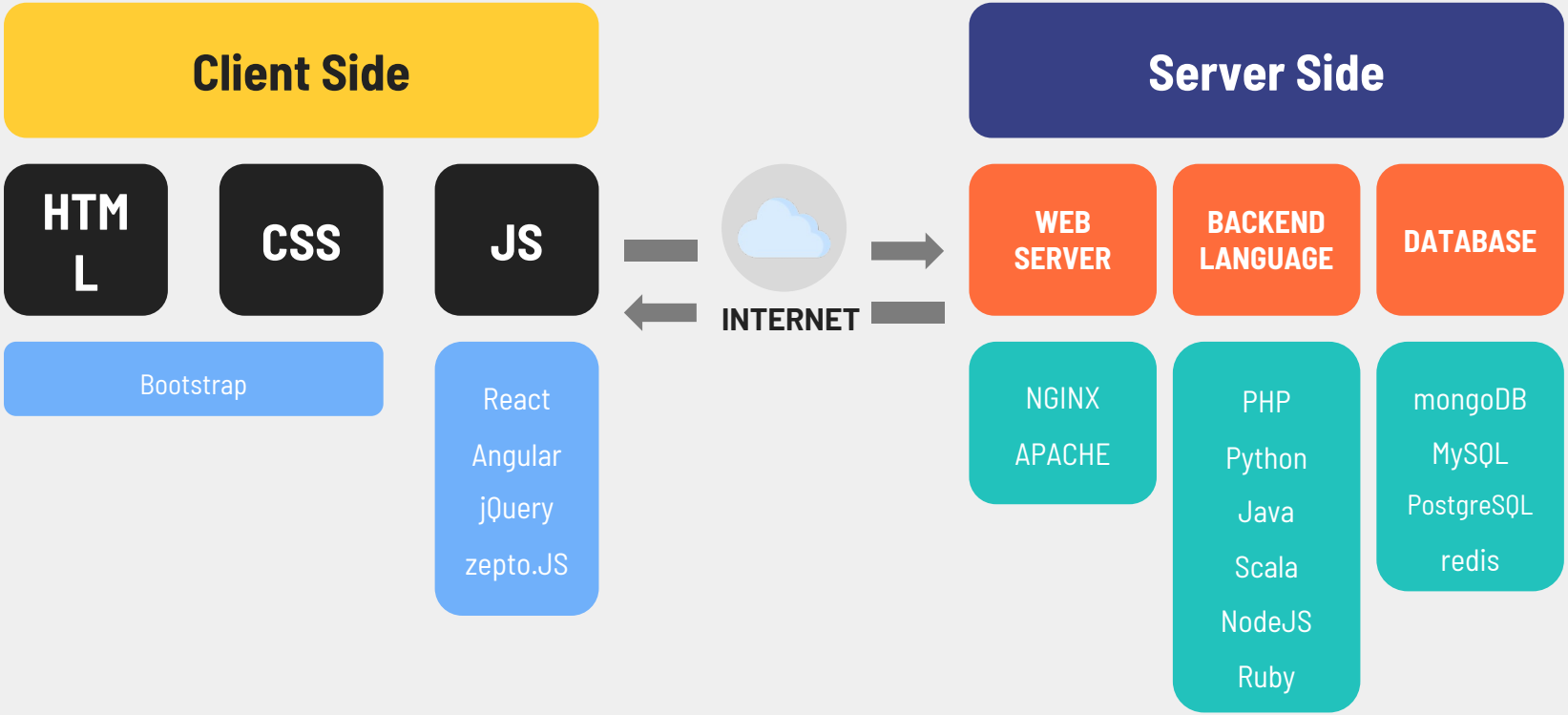
vs.

The Back-End/Server Side

- ➔ Database and frameworks, the operating systems that store key information
- ➔ What goes on behind the scenes
- ➔ Most of the code that exists is on the back-end and never seen by users
- ➔ The actual logic behind the application and the architecture of the system

Java, Python, Ruby, SQL, and more

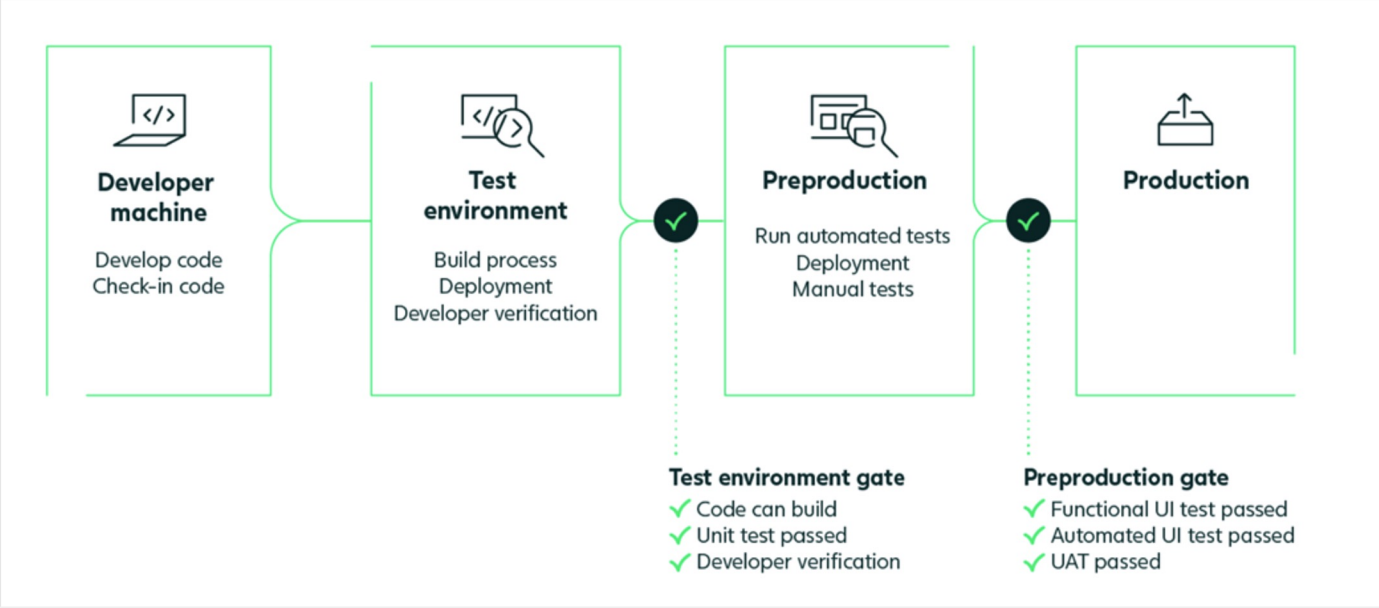
THE STACK



FRONT-END vs. BACK-END

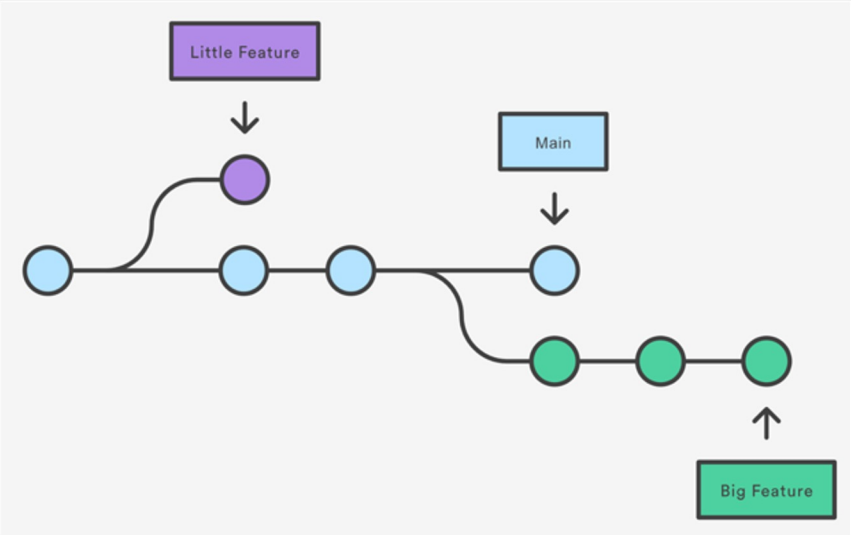
Deployment Pipeline

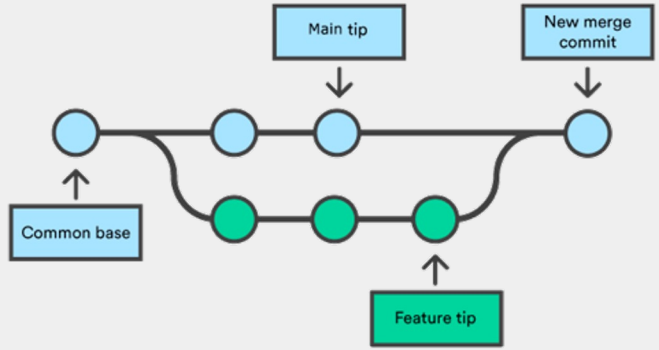
The process of **building, testing, and deploying** working code through progressively more realistic environments until it is “ready for use” by the customer.



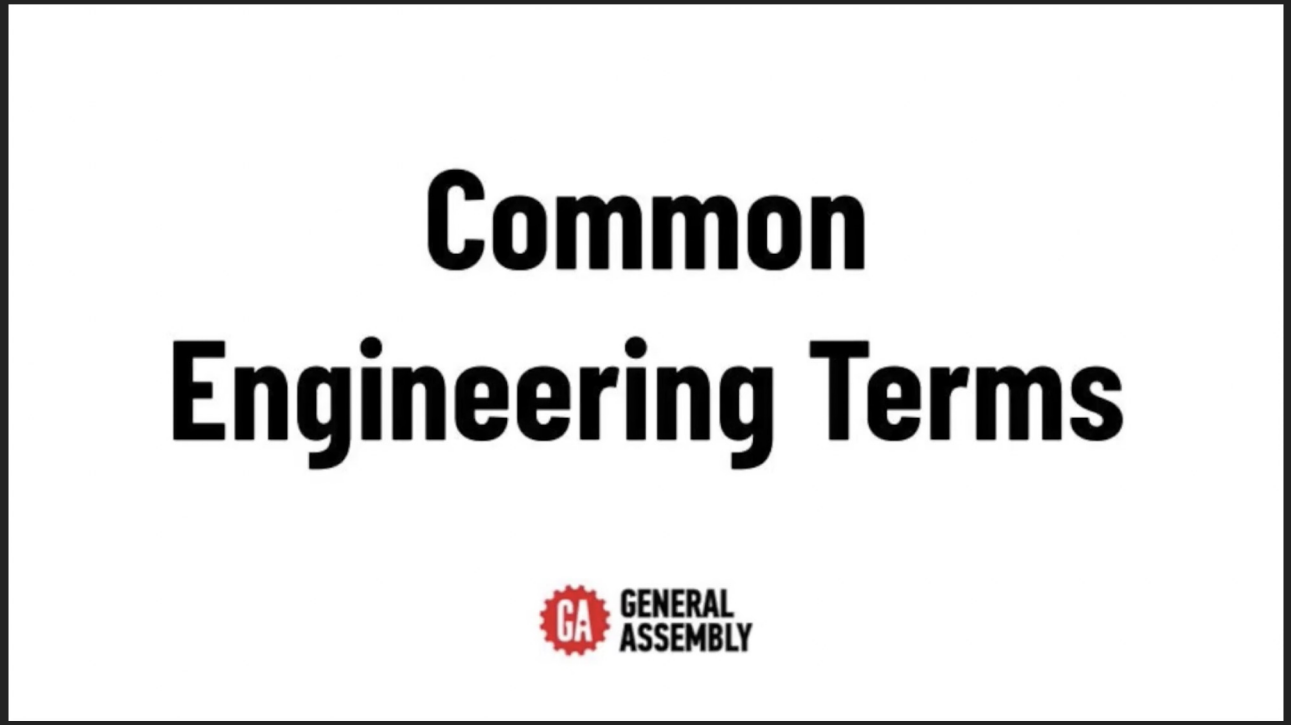
Git: Version control, Branching, Merging

The process of **managing the "Master" version of the codebase**, and the process of **creating and merging any "branches"** that split off from the Master.





COMMON ENGINEERING TERMS



https://generalassembly.wistia.com/medias/pttc8jqshh

Term	Definition
HTML	HTML stands for **H**yper **T**ext **M**arkup **L**anguage; the standard markup language for creating Web pages. HTML describes the structure of a Web page (W3 Schools).
CSS	CSS stands for **C**ascading **S**tyle Sheets; describes how HTML elements are to be displayed on screen, paper, or in other media (W3 Schools).
Javascript	The programming language of the Web; the world's most popular programming language (W3 Schools)
Python	Popular programming language that has a simple syntax similar to the English , allowing developers to write programs with fewer lines than some other programming languages
SQL	SQL stands for **S**tructured **Q**uery **L**anguage; lets you access, combine and manipulate databases.
Deployment	The process of sending new or updated code into production
The Cloud	Software and services that run on the internet, rather than on your computer

Term	Definition
API	A tool that allows two different programs to share data and interact
Git	A version control tool used for tracking changes during development
GitHub	A website for sharing and collaboration on software (and uses Git for version control)
DevOps	A discipline focused on making software development and deployment faster and easier
Production	The live version of a site that users see, not the version being built and tested by engineers
Mobile Web	Access of browser from device; sites are responsive (configured) to scale to size of device
iOS	Apple's mobile operating system
Android OS	Android's mobile operating system

SOLO EXERCISE: TECH TERMS RECAP

10 Minutes

Whew! We covered a lot of new technical terminology in this section.

On your own:

In Slack, write down one term or concept on which you need more clarity. Look at what others have written. **Can you explain what any of them are about? Do you need clarity on a term, too?**

Add a reaction: ➕ Need clarity ✅ I could explain it!

As a class:

We'll discuss and define some of the terms that are still unknown.

How do you learn your company's stack? Ask an Engineer.

BREAK TIME

COMMUNICATING WITH ENGINEERS

Working with Developers

Welcome + Warm-Up

Navigating the Tech Stack

Communicating with Engineers

Managing Last Minute Development Requests

Bring It Home

There are things that product may not be best positioned to foresee that engineering brings up, like scalability and system stability limitations, getting engineers involved early in the game doesn't hamper your progress but rather aids with them being in your story with you."

Anuja Chavan

BEST PRACTICES FOR COMMUNICATING WITH ENGINEERS

Ask for Technical Overviews when first joining as org

- ➔ If you're seeking a refresher, make an initial attempt to research and bring specific questions to the conversation.

Deliver Clear Specifications

- ➔ The more detail you can provide to regarding the specifics of the feature, the less requirement there will be for additional meetings or inaccurate builds.

BEST PRACTICES

Leave the timelines/deadlines to them

- ➔ Product defines the "What/Why" and Engineering is responsible for the "How Long/When?" **Do not provide an estimate without talking to your engineering team.**

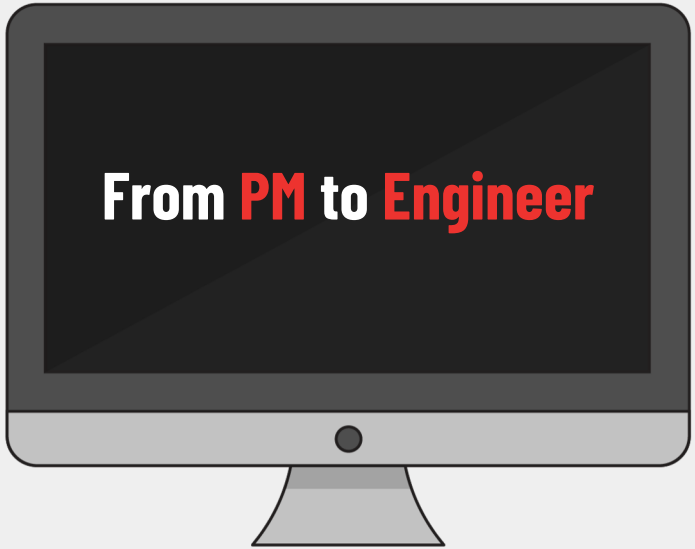
Include them in Conversations

- ➔ If your engineering team is willing to attend meetings, invite them to **key** conversations so that they can gain a full understanding of the work entailed.

COLLABORATION IS KEY

How do you determine technical constraints?

Work with your developers!



WHAT does the solution need to do?

WHY is this important to our users?

WHAT constraints do we have around costs and timelines?

HOW are customers dealing with this problem today?

Is there anything important to consider related to visual design?

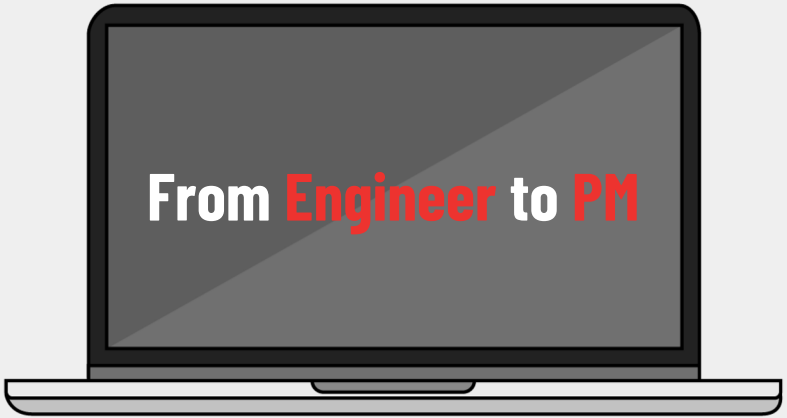
HOW will adding/changing this feature affect other features?

WHAT tools, infrastructure, and data do we already have that can be used?

HOW LONG will this take to build?

HOW MUCH will it cost?

WHAT is the least complex way of solving this problem?



IN A WORLD WITHOUT PLANNING AND COMMUNICATION...

Done as a “quick fix.”

Built now to solve an issue but may be paid for later with more complexity and issues.

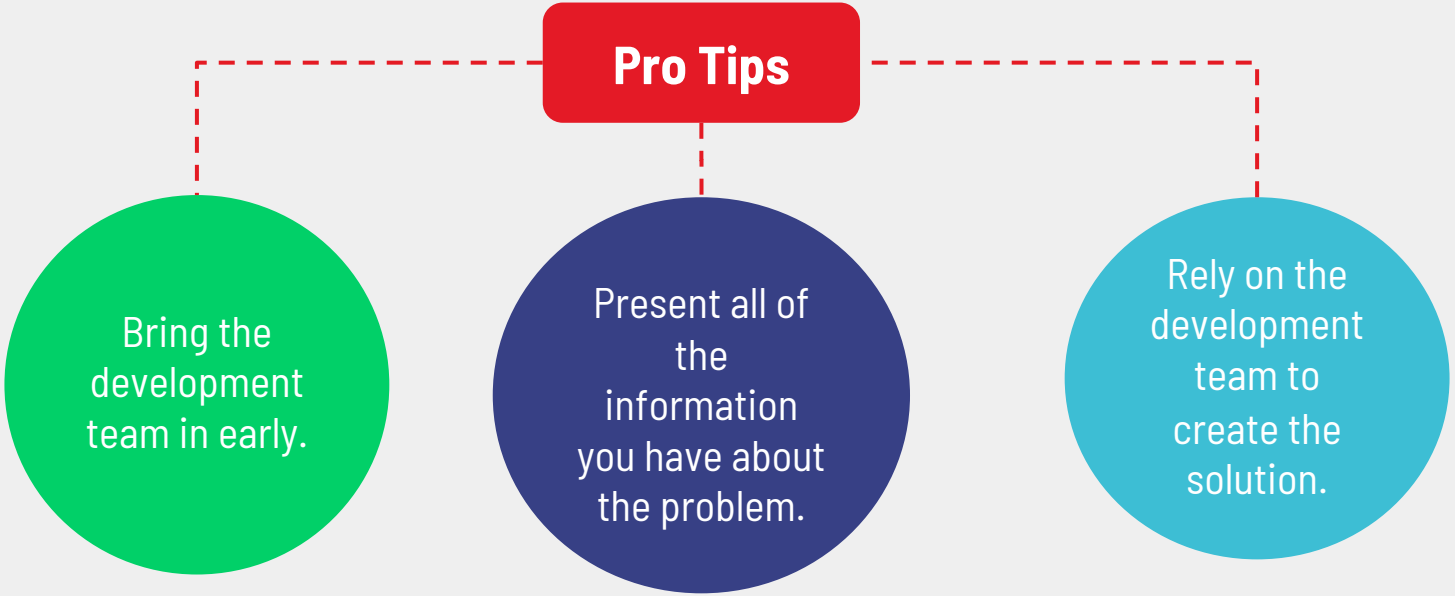
Make sure you’re making an informed decision — not all debt is all bad.

You could run into **technical debt.**

Customer’s View

Developer’s View

WORKING WITH DEVELOPERS - A REMINDER!



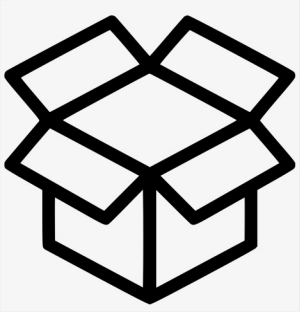
Remember: The developer is responsible for the "How" and the "When"

Arranging your Development buckets

Product Improvement (eg Features)

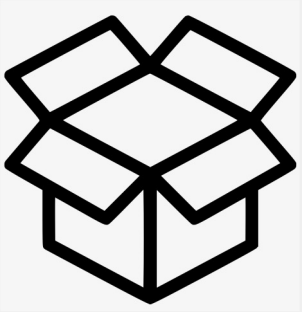
70%

Technical Excellence



20%

Bug Fixes



10%

MANAGING LAST MINUTE DEVELOPMENT REQUESTS

Working with Developers

Welcome + Warm-Up

Navigating the Tech Stack

Communicating with Engineers

Managing Last Minute Development Requests

Bring It Home

LAST MINUTE FEATURE REQUESTS

Most feature/product development occurs within a dedicated development cycle.

Oftentimes new/additional requests from stakeholders will come in with urgency.

The slightest change can have cascading effects.

Senior leadership can sometimes submit these requests as well.

HOW TO ADJUST WHEN NECESSARY

Sometimes those unplanned requests are a must-do. When that occurs, the team will need to pivot from their current work and focus on the new task at hand.

1

Assess the impact

- ➔ Delays to the current work
- ➔ Potential delays to future work
- ➔ Potential risks related to the adjustment

2

Communicate

Inform your development/design team, stakeholders and leadership.

3

Update

Your roadmap/feature tracking/specs to reflect the updated status of the work.

4

Track

The frequency of these requests as it may require an adjustment to your planning process.

5

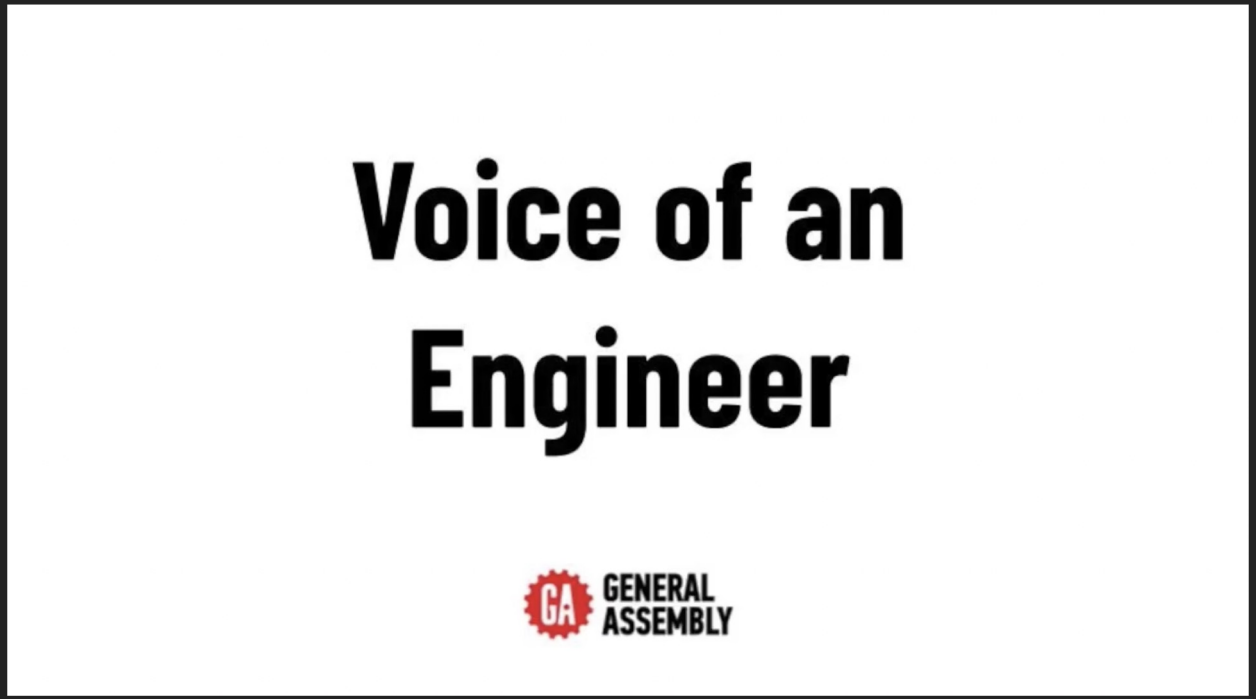
Listen

Carve out time to check in with your engineering/design team to acknowledge the impact the request has on their workflow.

VOP: VOICE OF THE PMs

https://generalassembly.wistia.com/medias/kb931n713b

VOE: VOICE OF ENGINEERS



https://generalassembly.wistia.com/medias/30hdkmywlr

DISCUSSION: LAST MINUTE REQUESTS

10 Minutes

Background:

Imagine that you are working with your development team on a key roadmap initiative. The team is about 1.5 weeks into development with an additional 2 weeks (estimated) left in development.

While the feature is straightforward the system is wrought with tech debt (almost every feature release is met with an outage - requiring extensive testing)

The team has built in an additional 1.5 weeks (included in the 2 weeks above) to address some of the major debt areas

You receive an email from senior leadership informing of a new feature that will need to go live ASAP and asks for a date.

How would you respond and what information would you gather to formulate your response?

BRING IT HOME

Working with Developers

Welcome + Warm-Up

Navigating the Tech Stack

Communicating with Engineers

Managing Last Minute Development Requests

Bring It Home

KEY TAKEAWAYS

Learn Your Tech Stack

Talk to your colleagues. Research and read documented overviews.

Communicate with Engineers

Strong PM to Engineering communication is essential and a key factor in building and scaling great products.

Avoid Last Minute Requests

Try to avoid last minute requests - if it's not possible, ensure that you communicate the impact on current and future work.

Additional Resources

Practice Again	Digging Deeper
Non-Technical PMs Working With Developers • So, You Want to Be a PM, but You Aren't Technical. Read This. • 10 Tips to Succeed as a "Non-Technical" PM • Top 5 Ways PMs Can Help Developer's Love Them	**Developers, APIs, & Technical Terms** • Tell Me About Your Technical Skills: A Non-Technical PM's Approach • Technology Skills for PMs • The Top 5 Technical Skills Every PM Should Know • DevOps Explained for PMs • APIs Explained for PMs • Building APIs Is Product Management | TIPM • What Is Technical Debt? | Definition and Examples


GA
GENERAL ASSEMBLY