



MGT 3745 B: communicating w/ markdown



# AGENDA

August 31, 2023

(103 days, 14 weeks, 5 days)

- key terms  
(markdown, version control)
- problem-solving  
methodologies
- js/markdown  
walkthrough
- assignments



# Key Terms

## Business (Computational Thinking) Communications

**Repository (repo)** - contains all of a project's files and each file's revision history

**Version control** - systems responsible for managing changes to computer programs, documents, large websites, or other collections of information (e.g., git)

**Markdown (\*.md)** - plain text files that contain a lightweight markup language for creating formatted text using a plain-text editor.

**README.md** - markdown file containing a summary of a repo/code to help others understand and navigate code. A README.md is typically included for every repository and contains a repository license, citation file, contribution guidelines, and a code of conduct, which communicates expectations for your project.

Source: <https://docs.github.com/en/repositories/creating-and-managing-repositories/about-repositories>

Source: [https://en.wikipedia.org/wiki/Version\\_control](https://en.wikipedia.org/wiki/Version_control)

Source: <https://en.wikipedia.org/wiki/Markdown>

Source: <https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/about-readmes>



# Why Use Markdown?

## What is the Curse of Knowledge?

"humans assume others have the same knowledge they do"

## What is a solution to "the curse"?

"...understand what your users want from your software and where they need help along the way"

- **User personas** - represent your ideal reader(s)
- **User stories** - describe a reader(s) intention
  - As a [type of user], I want [activity] so that I can [goal]
- **User journey maps** - a diagram that defines the user's task and visualizes the channels, steps, and experiences in a flow.
- **Friction Log** (aka pain points) - unofficial document where YOU try your software as an objective user would, log each step sequentially, and note the behavior you expect and the actual behavior of your software.

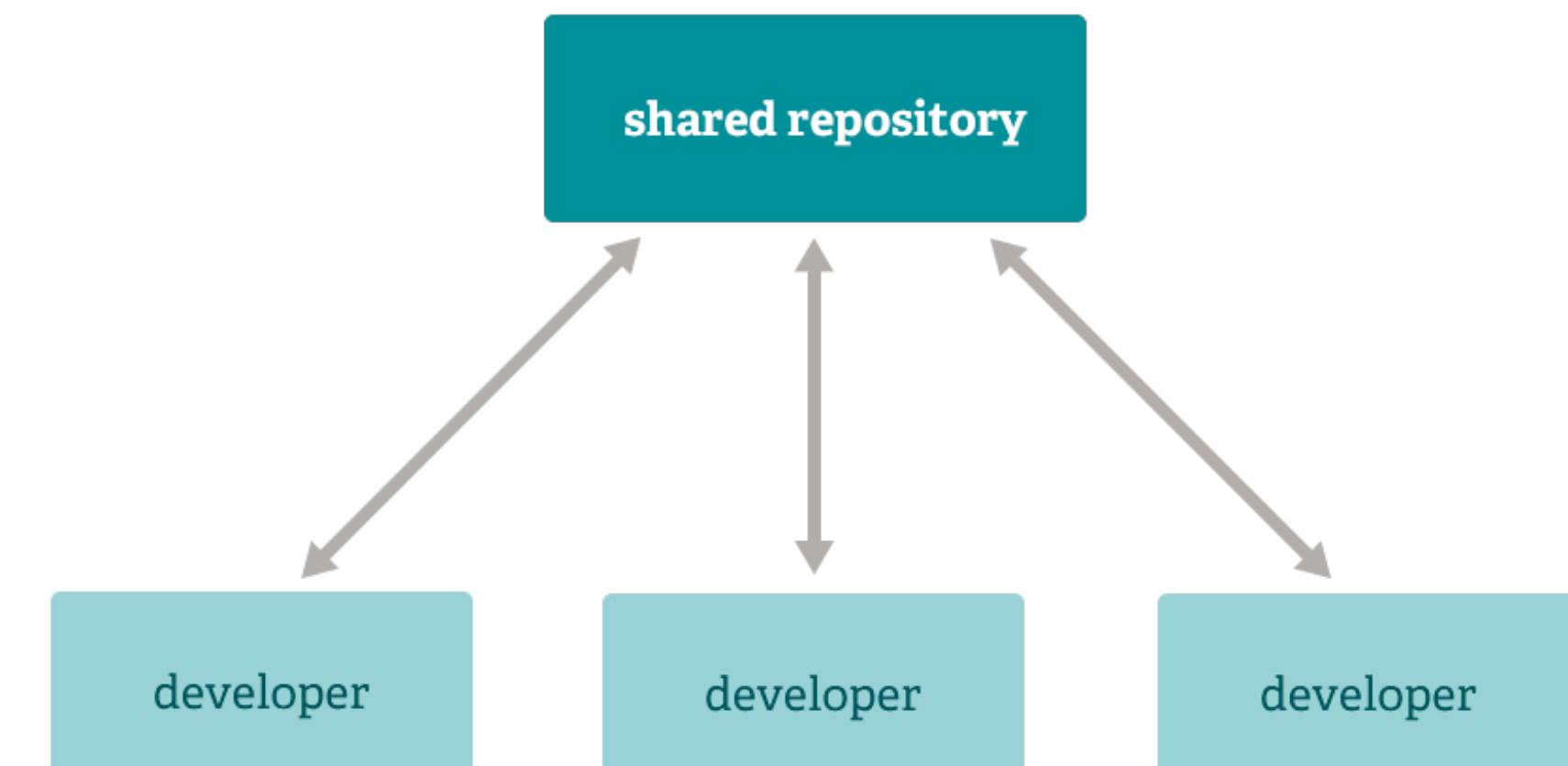
Source: [Docs for Devs \(Chpt 1\)](#)

# Version Control

**Git** is a free and open-source distributed version control system designed to handle everything from small to massive projects quickly and efficiently.

**GitHub** is a code hosting platform for collaboration and version control. GitHub is a social network that lets you (and others) work together on projects.

- ▶ long-term change history of every file
- ▶ branching and merging (keeps individual changes separate and allows for verification before merging back together)
- ▶ trace each change made to the software (blame)
- ▶ some of the largest entities in the world use GitHub to collaborate with the public via open source and hire resources
- ▶ shared/decentralized repo
- ▶ repos contain README.md files



Source: [What is version control](#)

Source: [About GitHub](#)

Source: <https://git-scm.com/about/>

# Key Terms

## Examples of Business Problems

**searches (e.g., filtering)** - processes and algorithms used to solve a search problem (finding "y" within "x")

**sorting** - ordering/categorizing data

**counting** - start/maintain a count of the number of iterations/cycles/instances

**lists** - start/maintain a structure of values

**personalization** - tailoring a service or a product to accommodate specific individuals, sometimes tied to groups or segments of individuals

**robustness** - ability of a computer system to cope with errors during execution and cope with erroneous input

Source: [https://en.wikipedia.org/wiki/Search\\_problem](https://en.wikipedia.org/wiki/Search_problem)

Source: [https://en.wikipedia.org/wiki/Random\\_number\\_generation](https://en.wikipedia.org/wiki/Random_number_generation)

Source: <https://en.wikipedia.org/wiki/Personalization>

Source: [https://en.wikipedia.org/wiki/Robustness\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Robustness_(computer_science))

# Key Terms

## Problem Solving Methodology

Create a **problem statement**

Use computational thinking as the process which **produces an algorithm/pseudocode** (plain language description of the steps) that generates the desired condition or result

Convert algorithms/pseudocode, or **encode**, into JavaScript

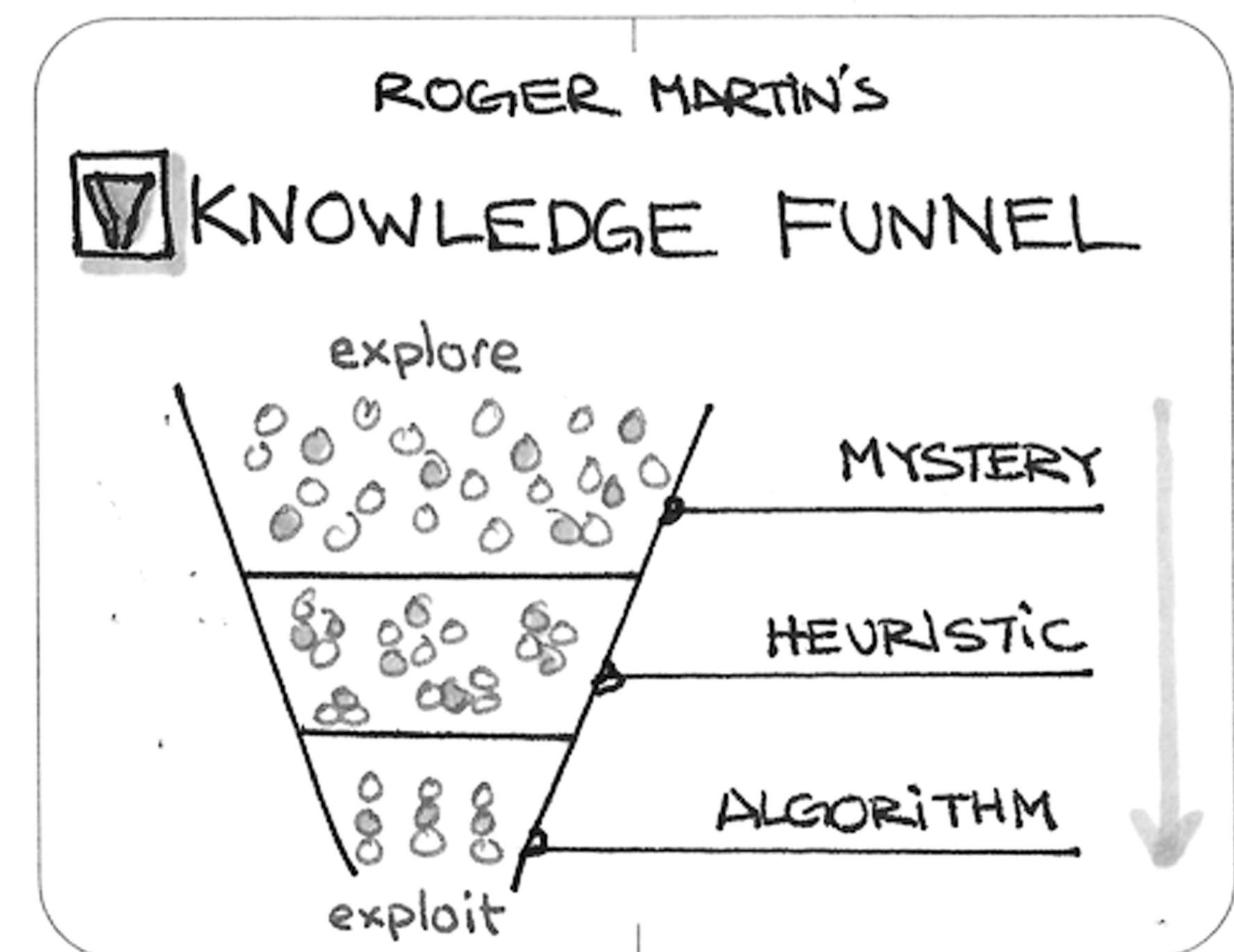
In-class exercise: <https://replit.com/@johnmcswain/MGT3745Fall2023Aug31>

Source: <https://en.wikipedia.org/wiki/Pseudocode>

Source: <https://rogerlmartin.com/lets-read/the-design-of-business>

sketchplanations.com

145





# Assignments

## Homework 1

<https://gatech.instructure.com/courses/347044/assignments/1497398>

## Homework 1 Discussion

[https://gatech.instructure.com/courses/347044/discussion\\_topics/1538581](https://gatech.instructure.com/courses/347044/discussion_topics/1538581)

## Homework 2

<https://gatech.instructure.com/courses/347044/assignments/1504834>

## Homework 2 Discussion

[https://gatech.instructure.com/courses/347044/discussion\\_topics/1551191](https://gatech.instructure.com/courses/347044/discussion_topics/1551191)



# Assignments

## Readings

[Markdown Basic Syntax](#)

[Docs for Developers \(Chpt 2-3\)](#)

[GitHub: About READMEs](#)

[JavaScript \(functions, objects, template literals\)](#)

## Supplemental

[The art of business communication \(Chapt 3, 5\)](#)

# Questions?