DON'T PANIC LABS

Opportunistic Active Learning

For Actual Intelligence

Bill Udell

Chief Operating Officer

Don’t Panic Labs

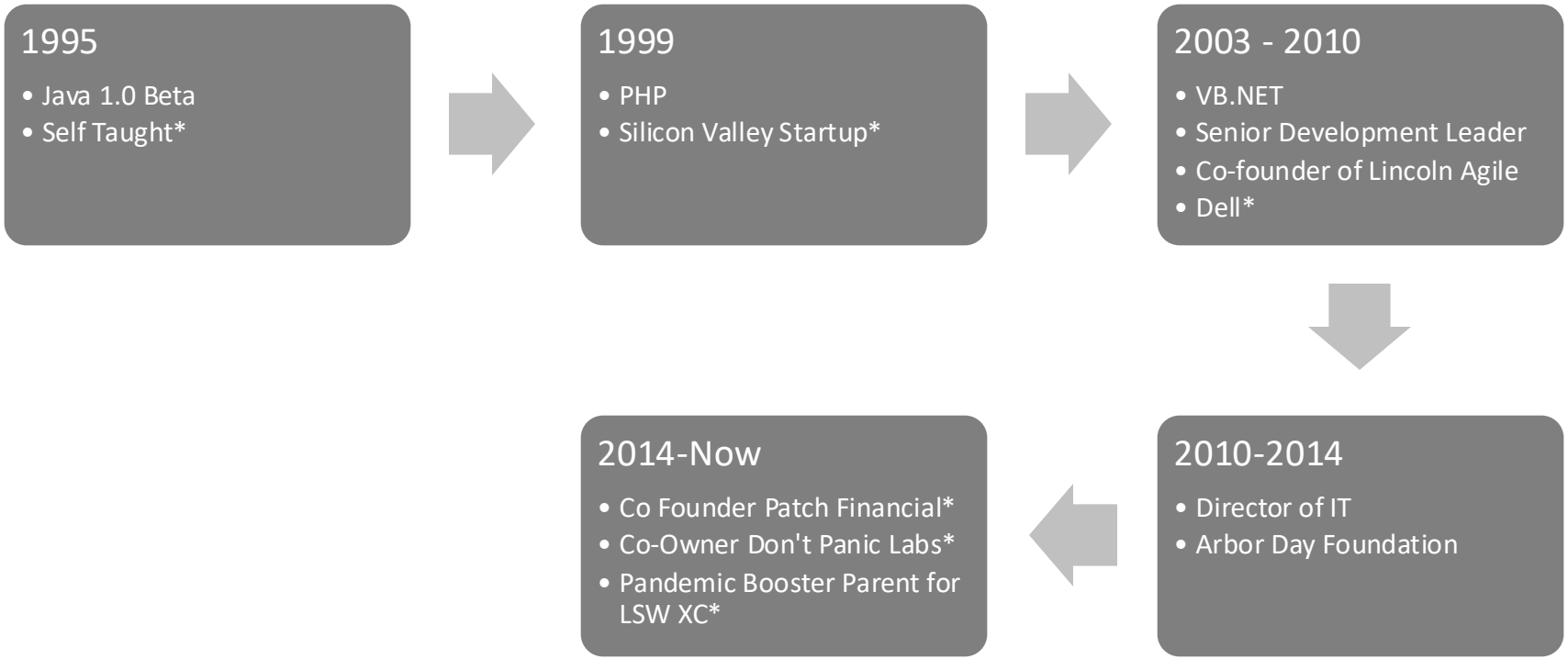
Questionable Credentials

BIO SCI

Computing Foundations

“Airport Bookstore Organizational Psychology”

Better “Credentials”



*If I didn’t asterisk these, I would likely tell a story about each one and we would never get to any content. If you want to hear stories find me at the after party. I’ll have drink tickets.

Milestones

2010

Nebraska Global is Founded

Don't Panic Labs is formed as software development arm

2014

Don't Panic Labs Begins External Product Development

Partners with local organizations on projects

2017

First Education Class Offered

Week-long programs to level up existing developers

2020

Partnership with Doane University

Nebraska DevLab launched to address tech talent shortage

2022

Don't Panic Labs Purchase

Buys full interest from Nebraska Global

Standish Group's 2020 CHAOS Report:

- 31% of projects were considered successful (delivered on time, on budget, and with required features and functions)
- 50% were challenged (late, over budget, or with fewer than the required features and functions)
- 19% failed (cancelled prior to completion or delivered and never used).

McKinsey & Company and University of Oxford

- Large IT projects (those with initial budgets exceeding $15 million) run 45% over budget and 7% over time, while delivering 56% less value than predicted.

IBM Study:

- An IBM study found that only 40% of projects meet schedule, budget, and quality goals.

…being a software engineer is not just about being a more highly skilled programmer

What we need to create our own opportunities for learning?

- Shared taxonomy and language of software product engineering competencies
- Strategies for Assessing competency attainment
- Tools to identify in-situ opportunities
- Activities and resources for improvement
- Thought leaders to fill the gaps

Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.

SOURCE: ISO/IEC/IEEE SYSTEMS AND SOFTWARE ENGINEERING VOCABULARY AND SWEBOK

DON'T PANIC LABS

YOUR VISION + OUR ENGINEERING

SWEBOK

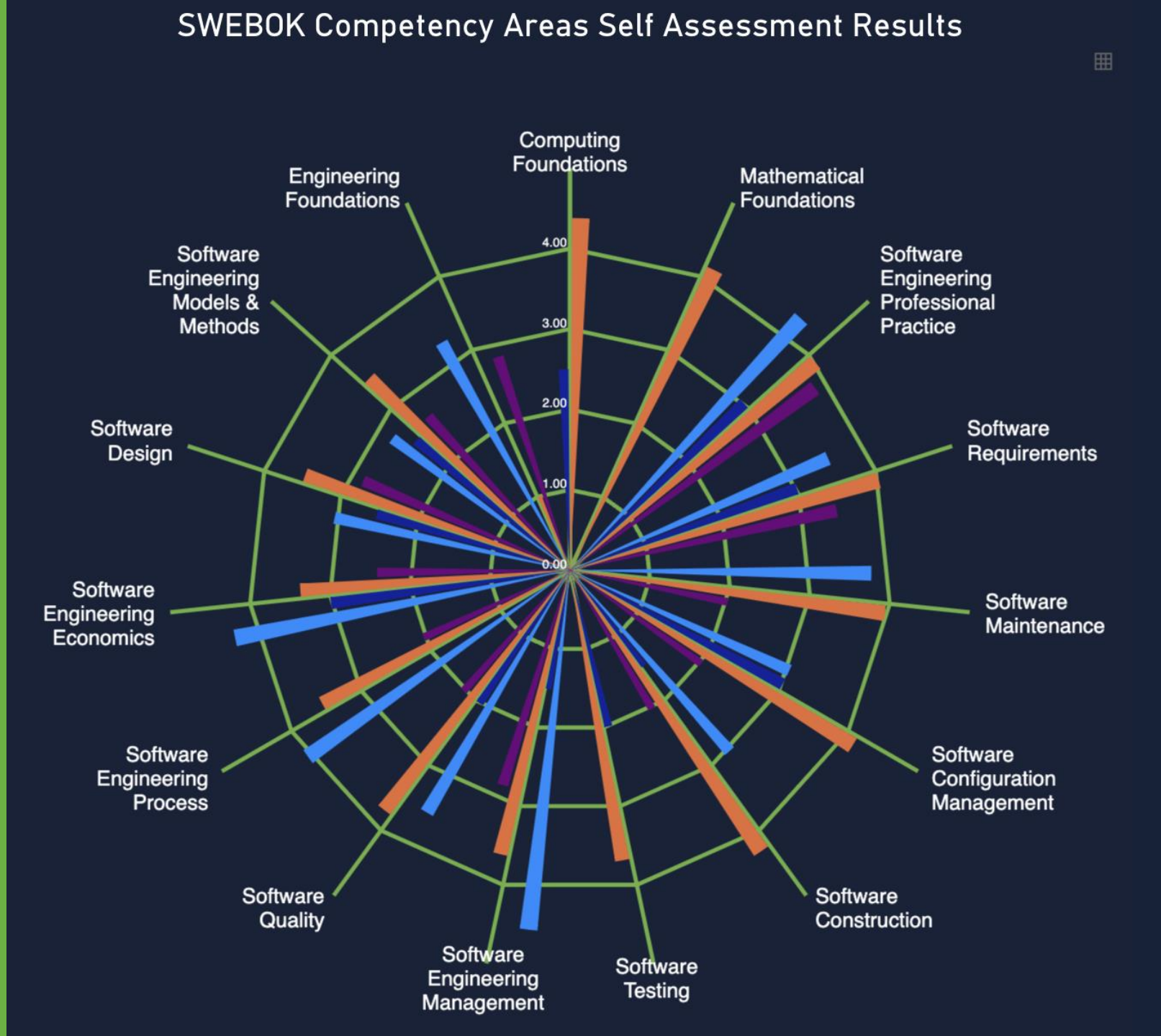
A taxonomy and basis for a professional standard of software engineering.

- **S**oftware **E**ngineering **B**ody **o**f **K**nowledge
  - IEEE
- Guide to the core areas of software engineering.
  - 15 knowledge areas (KAs)
  - Whole software life cycle, from requirements to maintenance.
    - References to standards, books, articles, and other sources.
- Descriptive and informative document, not a prescriptive or normative one.
- It does not tell software engineers how to do their work but what they should know and do.

DON'T PANIC LABS

YOUR VISION + OUR ENGINEERING

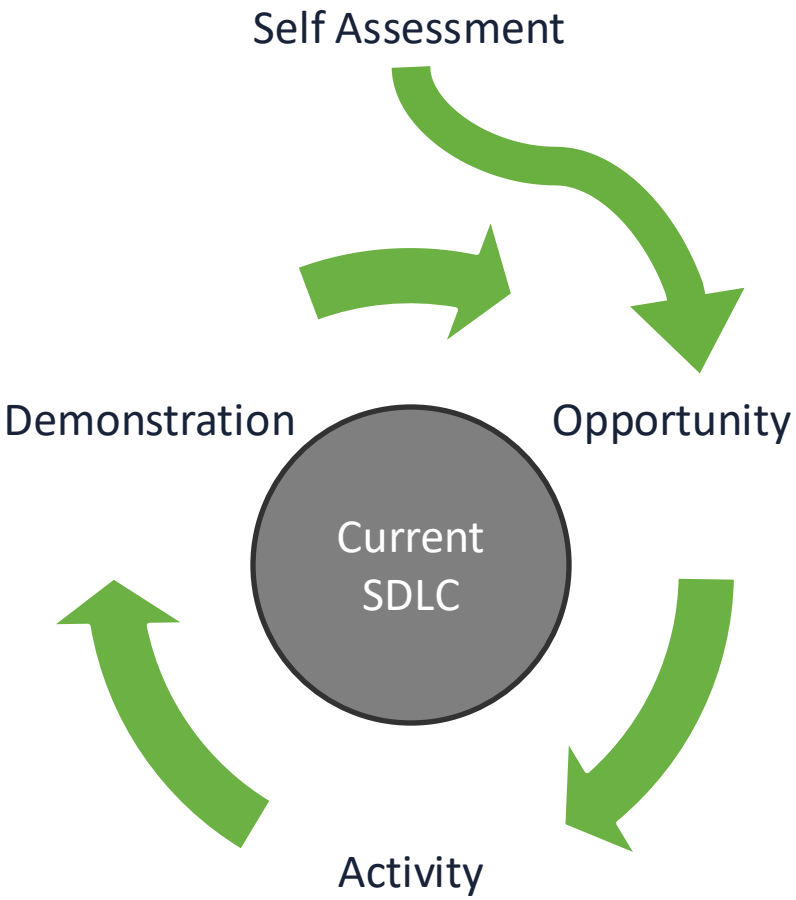
SWEBOK Competency Areas



DON'T PANIC LABS

YOUR VISION + OUR ENGINEERING

In-Situ Competency Development



DON'T PANIC LABS

YOUR VISION + OUR ENGINEERING

Self Assessment Example

Identifying Opportunities

- Use the critical thought activity you are already doing in your software development lifecycle
  - White papers
  - Estimates
  - Design
- Add an extra 30 to 60 seconds of critical thought to ask yourself a couple of questions.
  - When was the last time I _____?
  - When was the last time I rethought _____?
- **Create a 30-minute task to incrementally improve.**

“A good writer will almost always discover new things in the process of writing.” - Paul Graham, The Need to Read

“Writing exposes gaps in your knowledge and logic. It pushes you to articulate assumptions and consider counterarguments. Unclear writing is a sign of unclear thinking...” - *Adam Grant, Hidden Potential*

Seems true enough, but somewhat unscientific claim

It is nearly impossible to think about anything but the subject you are writing about while actively writing.

Writing for Critical Thought

- Pro Tip to be a “good writer” in this instance (you will not win a Pulitzer Prize)
  - Use disconnected self talk
    - Talk to yourself like a really good friend
      - Coach
      - Mentor
      - Teach
      - Encourage yourself
    - Literally say things like
      - “you may want to consider…”
      - “last time we did this you…”

DON'T PANIC LABS

YOUR VISION + OUR ENGINEERING

Critical Thought Tool: Whitepaper

We want to fill in something in all these areas

- For my work for this sprint
  - Impacts
    - Big picture thing we are helping to achieve
  - Outcomes
    - Measurable behavior changes we should expect
  - Outputs
    - Specific requirements
  - Activities
    - How we are going to do it
  - Resources
    - What we need to do it

Whitepaper Example

Bottom Menu Navbar Whitepaper

[redacted]

Created Date: 1/19/2023

Last Updated: 1/19/2023

Applicable Stories in the backlog:

Persona	Story	Version	Call Chain
Contractor	Given that I am logged in as a Contractor And I am on a page that the bottom menu navbar is displayed When I view the bottom menu navbar Then the following options/icons should be visible: • Inbox • Projects • Invoices • Archive • More		Front end See Ionic Tabs navigation
Owner	Given that I am logged in as an Owner And I am on a page that the bottom menu navbar is displayed When I view the bottom menu navbar Then the following options/icons should be visible: • Inbox • Projects • Create Project Request (large plus icon) • Archive • More		Front end See Ionic Tabs navigation The Create Project Request button should use the Ionic Floating Action Button

What is my understanding of the requirements? What is the primary goal or objective?

- The primary goal is to build out a navbar that supports Contractor and Owner roles

What are the secondary goals/objectives?

- Setup Ionic framework on the front end
- Scaffold pages that the navbar will link to, and set up the links to those pages

What are the known design constraints?

- Navbar should be displayed properly on mobile devices
- Displaying the navbar should be optional for each page
  - Ex: if you're in the middle of the sign-up process or the create project request process, the tabs should not be visible
  - They should be visible by default
- Navbar menu layout and links should be different for Owner and Contractor roles
  - Probably want to design it in a modular way in case additional roles are added

- Individual navbar menu buttons should be highlighted with the proper color if the user is currently viewing the page that is linked to by that button

What are the acceptance criteria for the solution (including non-functional)?

- Navbar menu is optional – individual pages should have the ability to toggle between displaying and hiding the menu
  - I'm thinking that this will be accomplished by having an **@Input** in the Tab navigation component and **@Outputs** in child components
- Navbar menu is modular – display of each individual link is controlled by the role of the current user
- Url is not impacted by existence of navbar – Ionic Tabs will be used for the implementation of the navbar menu, and Ionic Tabs come with their own routing. "Tabs" should not appear in the url, the actual destination should just be passed along
- Navbar menu buttons are highlighted when viewing the corresponding page
  - Apart from the floating action button used for the Create Project Request use case – on those pages, the menu won't be visible

What assumptions am I making about these requirements and the system?

- I am making some assumptions about the best way to design this component, please see the **tradeoffs** section for discussion about the design
- Assuming that the application's appearance in a desktop view is not particularly important
  - Not that it will be disregarded, but simply that it's not the priority

What are the unknowns? (i.e., what information is not currently available?)

- I'm not aware of unknowns currently as the design is given clearly in Figma and the implementation is mostly straightforward

What are the tradeoffs I am deliberately making?

- I am deciding to have every page routed through the Tabs navigation, rather than having some pages routed through the Tabs and others not be, as I believe this simplifies the routing overall
  - This is part of the more specific decision to allow individual pages to decide whether the navbar menu should be displayed, rather than have the Tabs page be the authoritative source on whether to display itself on a page
  - A corollary decision is that the Tab navigation will be displayed by default, employing an opt-out approach rather than an opt-in approach
  - The tradeoff being made is that it may slightly increase the complexity of any component/page that wishes not to display the Tabs navbar menu, as well as potentially increase the complexity of any page that is concerned with routing between the Tabs navbar component and the child component
    - Ex: if a component located through Tabs/Project/Create wishes not to display the navbar, then (assuming I understand this correctly) the Project component will be involved in communicating that wish to the Tabs component from the Create component.

- The increase in complexity in the components between the Tabs and the target components may be mitigated through the use of an external service to handle this however

What areas/features are most likely to change over time?

- There could potentially be more roles added to the future. If a new role is added, the navbar would need to be updated to expect this role and display the correct icons/links

What are anticipated/possible failure scenarios and how should they be handled?

- A button links to a page that doesn't exist
  - This issue requires the intervention of a developer. Some options to fix could be:
    - Manually update the link – remove it or modify it to point to an existing page
    - Create the necessary page

Are there any special considerations related to security?

- Security is minimally related through the use of roles to determine which buttons / links should be visible, but it is not specially considered.

What are the existing areas impacted?

- There are no existing areas yet

How will the design/architecture need to change?

- Potentially adding a new utility service to manage communications between the navbar component and other components that wish to hide the navbar

How can I encapsulate current and future change?

- Encapsulate changes to the navbar buttons by implementing the display of each button in a modular way, controlled by role
- Encapsulate changes to whether the navbar should be displayed on a particular page by allowing that page to explicitly control whether it should be displayed
  - If a page wishes to hide the navbar, the only changes required will occur in that component and not in this navbar component

How should I test and validate the system (including what regression testing should I do)?

- Manual testing

What risks am I aware of?

- Not concerned about uncertainty, estimates, requirements ambiguity, etc

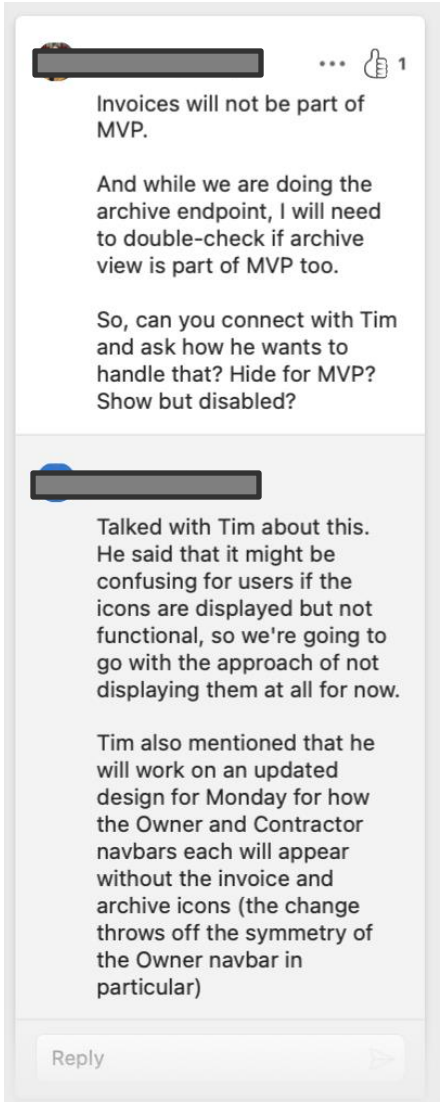
What concerns do I have?

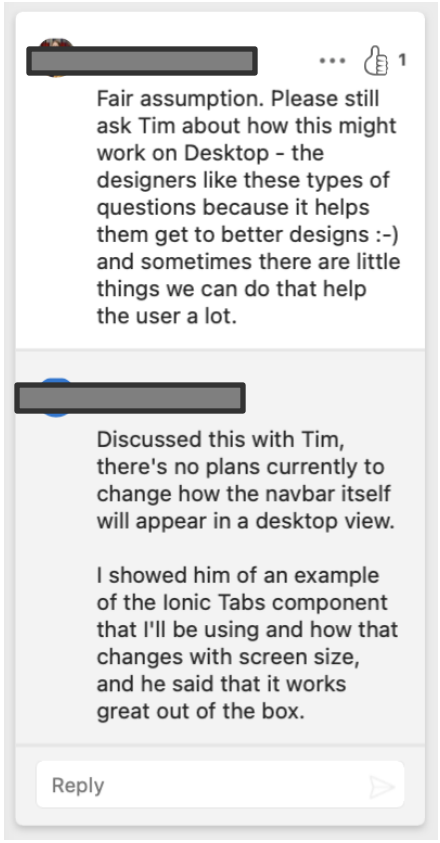
- No additional concerns

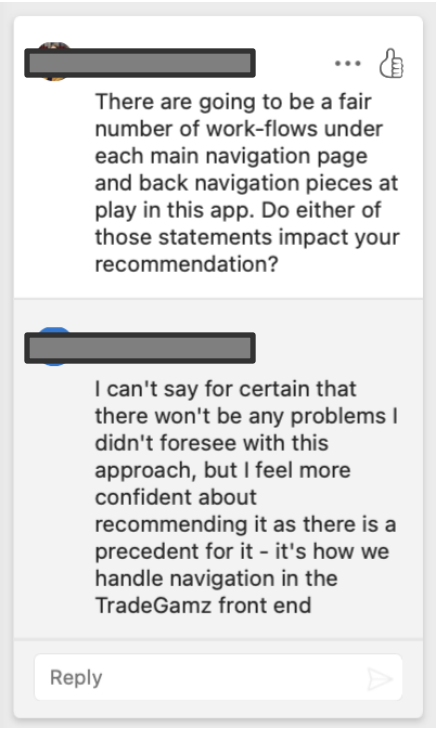
What are the specific steps to implement and what is their level of effort?

- Step 1: 0.2 - 0.4
  - Add Tabs component with navigation following the Ionic guide
  - scaffold pages that will be linked to by this navbar

Whitepaper Example







Likely Competency Growth Areas

SWEBOK Strength Areas to Focus for [redacted]

- Software Testing
- Software Design
- Software Requirements
- Software Engineering Models & Methods
- Computing Foundations
- Mathematical Foundations

“The way you like to learn is what makes you comfortable, but it isn’t necessarily how you learn best.”

— **Adam M. Grant,**

Learning from Resources

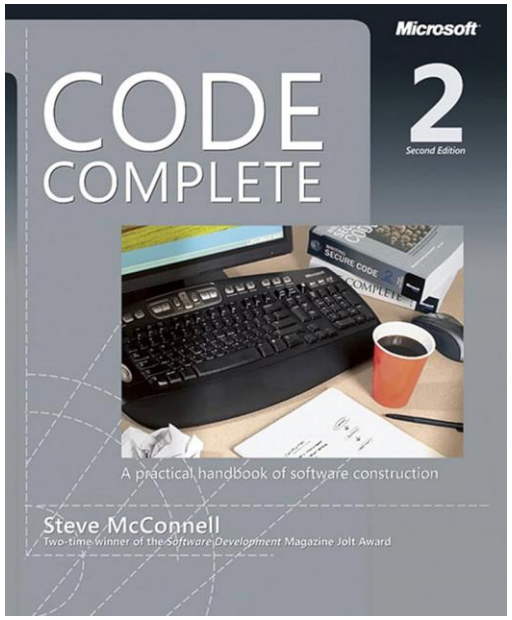
- Make yourself uncomfortable.
- When you read something, write about it.

DON'T PANIC LABS

YOUR VISION + OUR ENGINEERING

Competency Learning Activity Example

- Read: Code Complete
  - Chapter 3.4 Handling Requirements Changes During Construction
- Do some Gherkin of what we learned from Tim
- Write a blog post.



Activities and Resources

- SWEBOK itself is a great resource for books and websites
- Internal knowledge bases
- Leaders and Peers

"Code Complete" by Steve McConnell - Considered a must-read for software developers, this book provides comprehensive coverage of software construction and is frequently cited in the SWEBOK for its authoritative insights on best practices in coding and software design.

"Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides (the Gang of Four) - This book is pivotal in understanding software design and architecture, offering a catalog of proven solutions to common software design problems.

"Software Engineering: A Practitioner's Approach" by Roger S. Pressman - This textbook covers a broad spectrum of software engineering principles and practices, aligning well with the various knowledge areas outlined in SWEBOK.

"Refactoring: Improving the Design of Existing Code" by Martin Fowler - Essential for understanding how to improve code quality and maintainability, this book provides practical techniques for refactoring code, which is a significant aspect of software maintenance and quality assurance.

"The Mythical Man-Month: Essays on Software Engineering" by Frederick P. Brooks Jr. - This collection of essays offers timeless insights into project management and software engineering practices, reinforcing the importance of proper planning and estimation in software projects.

We need to have better conversations.

You have a HUGE role to play in that.

A Call for Leadership

- Software Construction is well represented in online and at conferences
- Underrepresented Areas
  - Software Engineering Foundations
  - Software Engineering Economics
  - Software Quality
  - Software Ethics

Next Steps

- Learn more about the SWEBOK
- Implement an opportunistic active learning process for yourself and your team
- Become a thought leader

- Questions
  - Visit our Booth
  - Visit our website https://dontpaniclabs.com