re= research

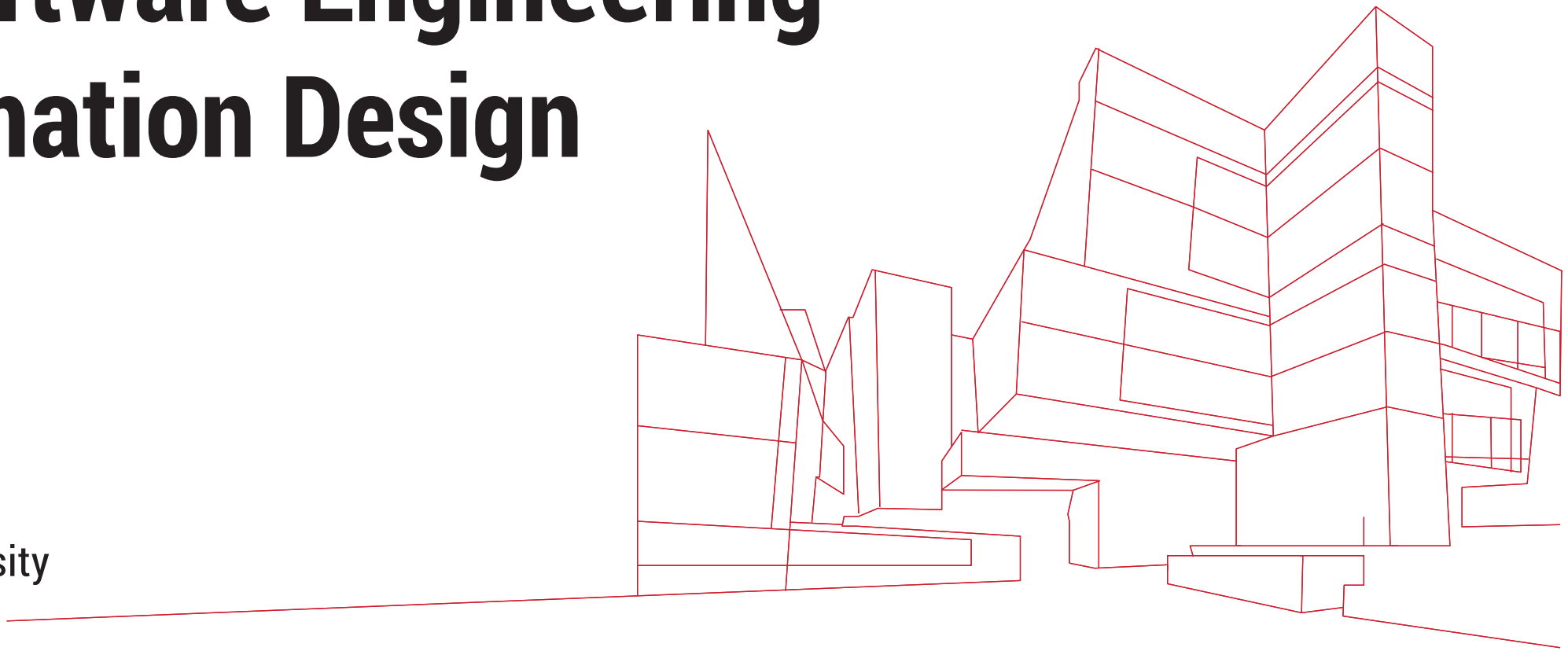
iasdr 2017

University of CINCINNATI

D | A | A | P

Columbus, Ohio | U.S.A.

From Software Engineering to Information Design



Yvette Shen

Assistant Professor
Department of Design
The Ohio State University

INTRODUCTION.

Given the vast varieties of practices and knowledge base required from information designers, it is crucial for them to look outside of the traditional visual design fields and seek diversities for better research and creation methods.

Software Engineering

Turning specifications into a working software system. It covers architectural views, as well as detailed component and algorithm implementation issues (Laplante, 2007)

Information Design

Using visual solutions to define, explain, or order contents, simplify complexity, relate the seemly unrelated, discover what has been hidden, create insights (Cairo, 2013)

technical solution →

BACKGROUND.

Multidisciplinary nature of solving contemporary complexity in all design forms.

vast varieties of practices and knowledge-bases required esp. for information designers

solving problems
design patterns
design models

precision, innovation, creativity, organization, rigorous logic

BACKGROUND.

Sample established design method and creative process often followed by Information Designers

Goal-Directed Design by Cooper consultancy

human-centered products and service designs that focus on achieving a series of goals.

Atomic Design by Brad Frost

break down user interfaces (responsive) into their basic components so that they can be reused throughout the system.

BACKGROUND.

Information Design rests on a large range of research foundations.

psychology, communication studies, sociology, linguistics, statistics, computer science, cognition science, and various design focus areas.

logical + analytical

creative + intuitive

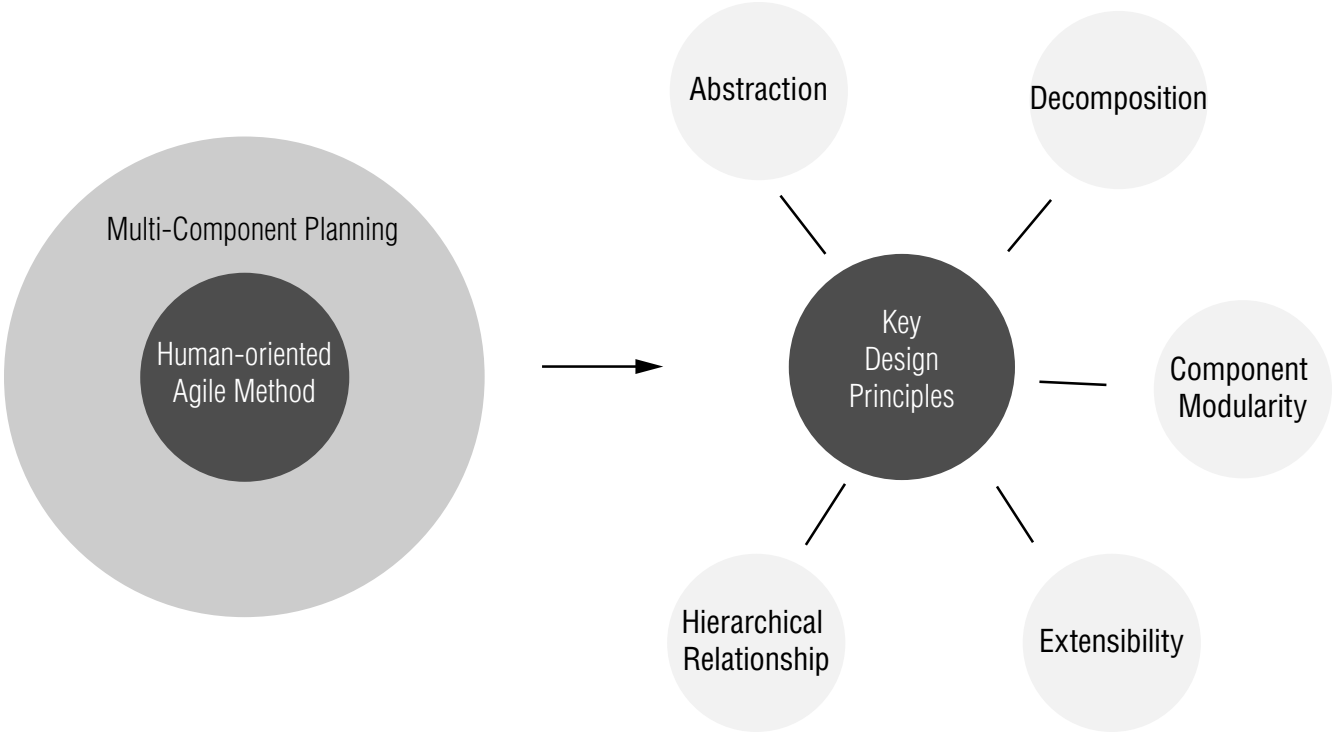
METHOD.

Design Framework Comparisons

Units of Analysis

Terminology Explanations

Forms of Communication



TAKEAWAYS.

Design Process

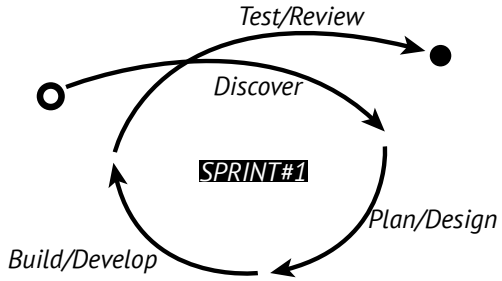
Agile and Sprints

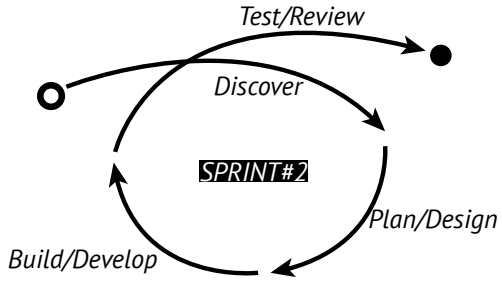
software engineering

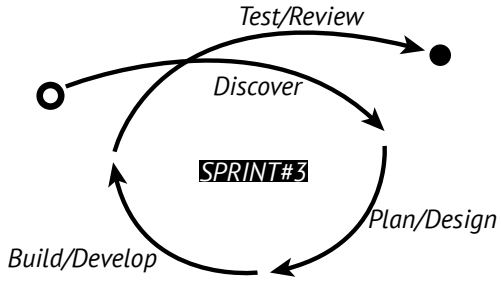
divide time into iterative and incremental cycles known as “sprints” to address needs.
customer satisfaction.
adaptive planning.
unit prototyping.
collaborative work environment.

information design

when elements of novelty are involves, such as evolving requirements, changing priorities, flexible deliverables.







TAKEAWAYS.

Design Principles

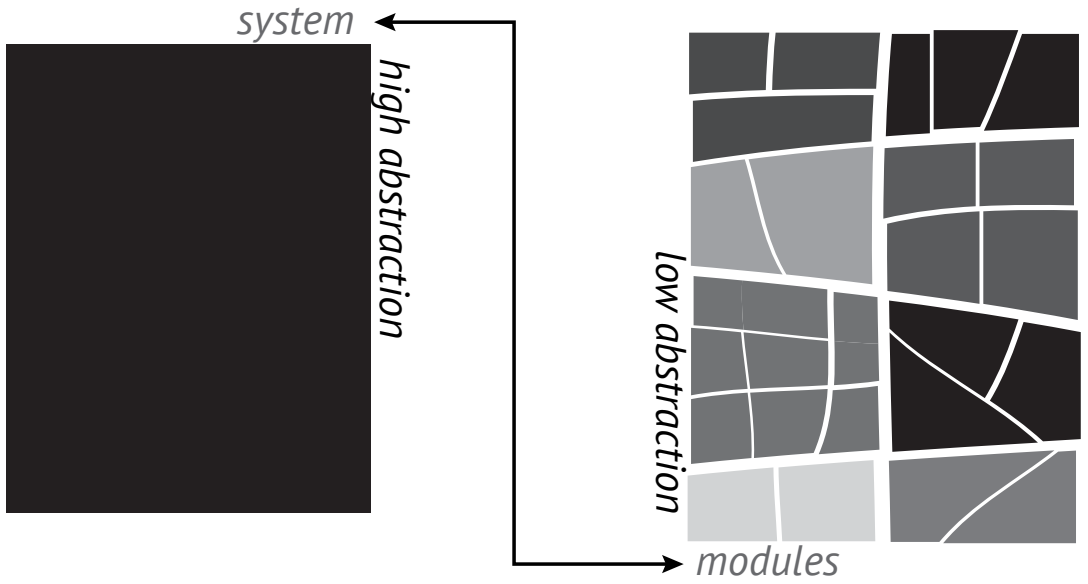
Abstraction

software engineering

manages complexity by emphasizing essential characteristics and suppressing implementation details. it allows postponement of certain design decisions that occur at various levels of analysis.

information design

use *abstraction* to filter and organize information. It can be applied to judge whether the collected information is relevant, useful, and explicit to the content and the visual form.



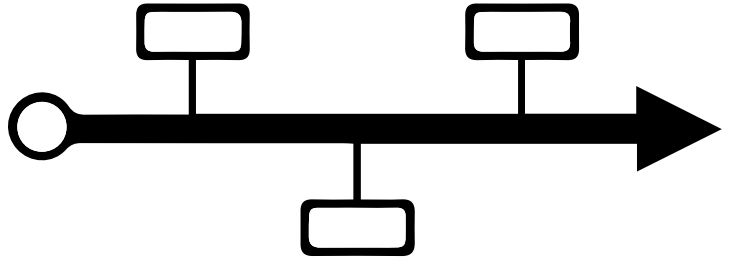
TAKEAWAYS.

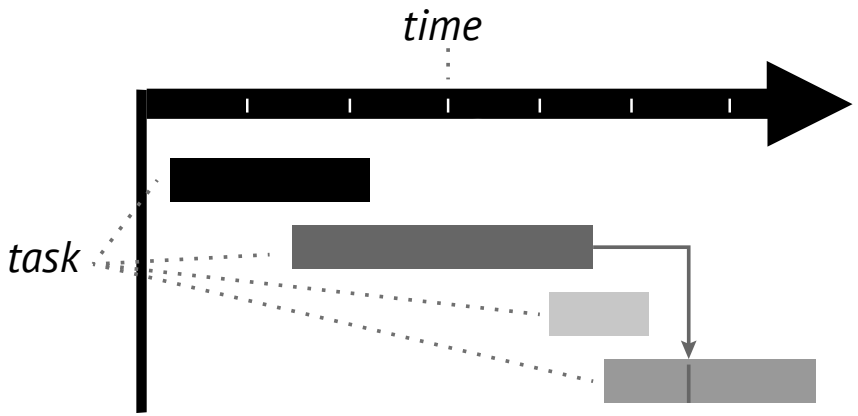
Design Principles

Decomposition

software engineering	information design
components within the large piece is determined using a design paradigm, such as functional, structured, object-oriented, etc.	dissect a complex problem into smaller building blocks by viewing interrelationships rather than the linear chains, and to recognize processes of change rather than snapshots

cause and effect
facts and comparisons
data and analysis





TAKEAWAYS.

Design Principles

Component Modularity

software engineering

a *modular* ***system*** is one that's structured into identifiable abstractions called *components*. a *component* can be a software entity that represents an abstraction, or it can be a unit of code that encapsulates data or has identifiable boundaries.

cohesion is one of the key concepts that are stressed in a component-based software system. It is the degree to which the components belong together within a module.

information design

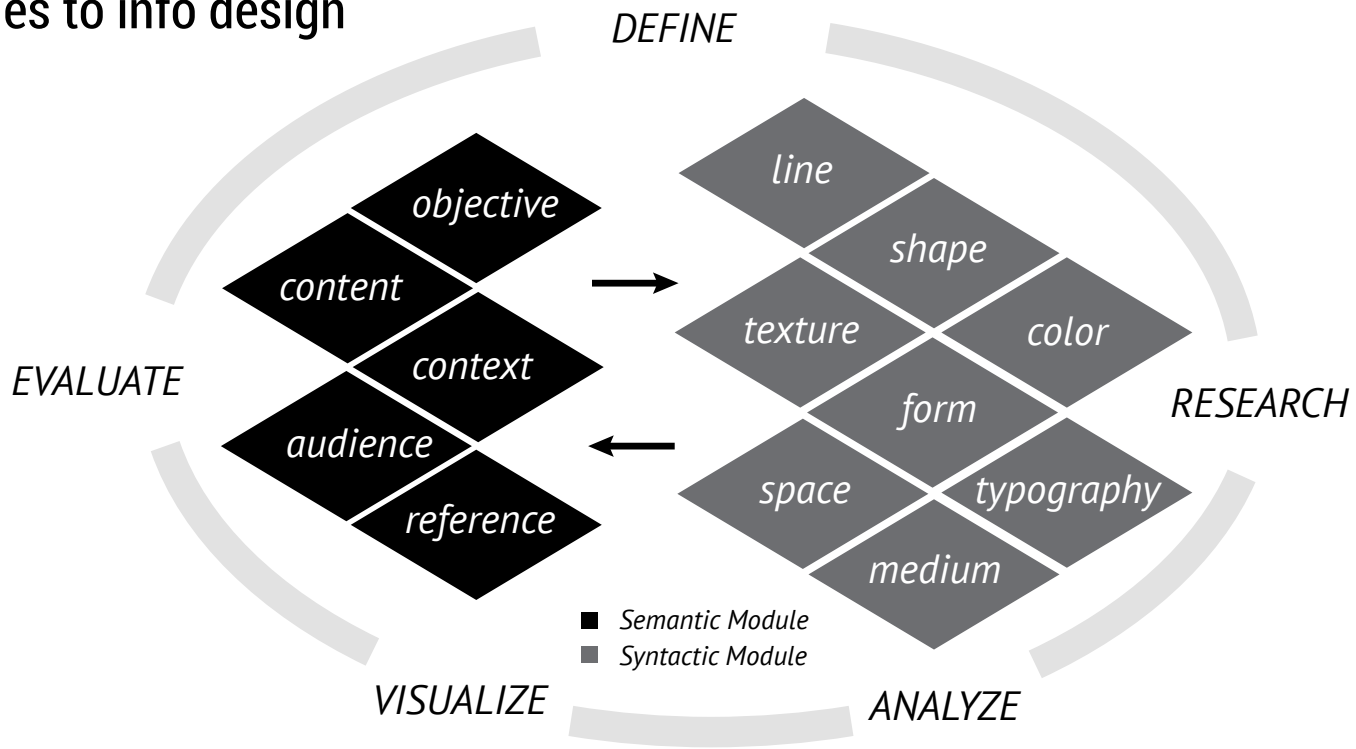
non-linear and recurrent matrix system.
the cycle that involves defining, researching, analyzing, visualizing, and evaluation are iterative, but also flexible and connected.
the modules can be categorized as semantic or syntactic

TAKEAWAYS.

Design Principles

Component Modularity

applies to info design



the connection between the variables forms a systematic approach that results in a framework of the semantic components to be expressed through a particular set of syntactic components.

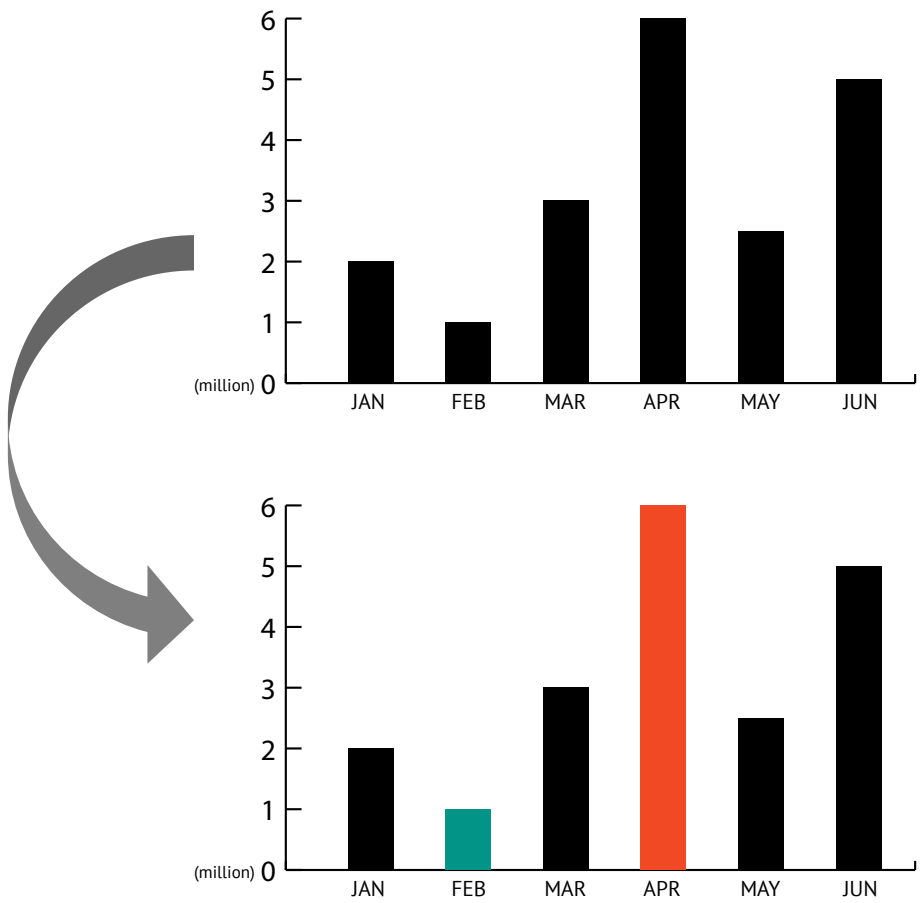
TAKEAWAYS.

Design Principles

Component Modularity

the individual component has its own purpose, yet they rely on each other to function as a whole.

a modification of one component may affect the rest.



TAKEAWAYS.

Design Principles

Extensibility

software engineering	information design
ability to take future growth into consideration. when new functionalities or requirements are extended, the existing structure and data flow are minimally or not affected.	flexible designs: various media outlets. visual framework consistency with dynamic data system. sustainable design solutions.

IN PROGRESS.

Applying the frameworks: case studies using
abstraction, decomposition, component modularity
in information and data visualization.

How a diversity of visual representations can give different meanings and interpretations to a single collection of data?

What variables or attributes in the dataset affect the choice of the visualization forms?

CONCLUSIONS.

Identifies design principles in software engineering design that resonate with information design.

The issues include the multi-component planning approach, the human-oriented agile method, and software design principles such as abstraction, decomposition, component modularity, hierarchical relationship, and extensibility.

intends to encourage more conceptual-based conversations between information design and other disciplines, especially in the field of science and technology.

REFERENCES.

Beck, K., & Andres, C. (2015). *Extreme programming explained: Second edition, embrace change*. Boston: Addison-Wesley.

Brown, T. (n.d.). Design Thinking. Retrieved March 17, 2017, from http://www.ideou.com/pages/design-thinking

Buschmann, F., Kircher, M., & Schmidt, D. C. (2008). *Pattern oriented software architecture*. Chichester: Wiley.

Cairo, A. (2013). *The functional art an introduction to information graphics and visualization*. Berkeley: New Riders.

Frost, B. (n.d.). Atomic Design Methodology. Retrieved March 17, 2017, from http://atomicdesign.bradfrost.com/chapter-2/

Goodwin, K. (2009). *Designing for the digital age: how to create human-centered products and services*. Indianapolis: Wiley Publishing.

Hansen, Y. (1999). Visualization for Thinking, Planning, and Problem Solving In *Information Design* (pp. 193-220). MIT Press.

Horn, R. E. (1999). Information Design: An Old Hag in Fashionable Clothes? In *Information Design* (pp. 15-34). MIT Press.

IEEE Standard Glossary of Software Engineering Terminology. (n.d.). doi:10.1109/ieeestd.1990.101064

James, M. (2015). Scrum Methodology. Retrieved August 24, 2017, from http://scrummethodology.com/

Laplante, P. (2007). What Every Engineer Should Know about Software Engineering. *What Every Engineer Should Know*. doi:10.1201/9781420006742

Manifesto for Agile Software Development. (2001). Retrieved March 17, 2017, from http://agilemanifesto.org/

Miller, M., & Boix-Mansilla, V. (2004). Thinking Across Perspectives and Disciplines. *GoodWork Project Report Series,27*.

O'Grady, J. V., & O'Grady, K. V. (2008). *The information design handbook*. Cincinnati, OH: How Books.

Passini, R. (1999). Sign-Posting Information Design in *Information Design* (pp. 83-98). MIT Press.

Pressman, R. (2014). *Software engineering*. New York: Mcgraw-Hill.

Schmidt, D. (2003) Software Design Principles and Guidelines Design | FlipHTML5. Retrieved March 17, 2017, from http://fliphtml5.com/bqjb/mqoa

Shedroff, N. (1999). Information Interaction Design: A Unified Field Theory of Design in *Information Design* (pp. 267-292). MIT Press.

Stinson, L. (2017, March 15). John Maeda: If You Want to Survive in Design, You Better Learn to Code. Retrieved March 17, 2017, from https://www.wired.com/2017/03/john-maeda-want-survive-design-better-learn-code

THANK YOU

Yvette Shen

shen.1049@osu.edu