CMPUT 401
Software Process and Product Management

Ildar Akhmetov

ildar@ualberta.ca

Department of Computing Science

University of Alberta

Client Needs and Software Requirements

Fall 2020

What is a requirement?

A specific description of your client’s needs.

Requirements Activities

Eliciting	Eliciting requirements
Expressing	Expressing requirements
Prioritizing	Prioritizing requirements
Analyzing	Analyzing requirements
Managing	Managing requirements

1. Eliciting Requirements

“Wants”

“Needs”

2. Expressing Requirements

- Use cases
- Storyboards
- User stories
- Story maps

3. Prioritizing Requirements

MoSCoW

Must be done

Should be done

Could be done

Would like but **won’t** get

4. Analyzing Requirements

Examining the listed requirements of a project to ensure they are clear, complete, and consistent

5. Managing Requirements

- Keeping track of priorities, analyses, and changes in requirements
- Ensuring that the identified requirements are central to the many processes of product creation, including coding, testing and change logs

Can influence the project	**Most important**	Add context
Business requirements	**User requirements**	External interfaces
Business rules	**Functional requirements**	Physical product settings
	Non-functional requirements	Development constraints

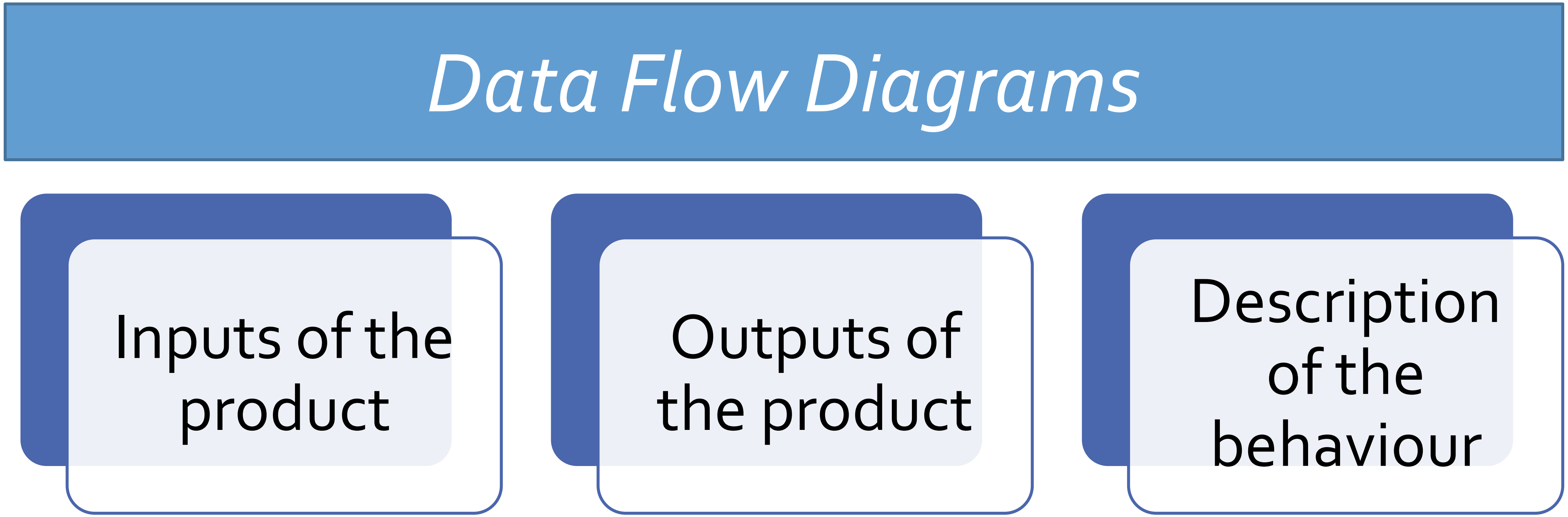
Types of Requirements

User Requirements

What the product can do for the user

- Use cases
- User stories
- Story maps
- Storyboards
- Scenarios

Functional Requirements



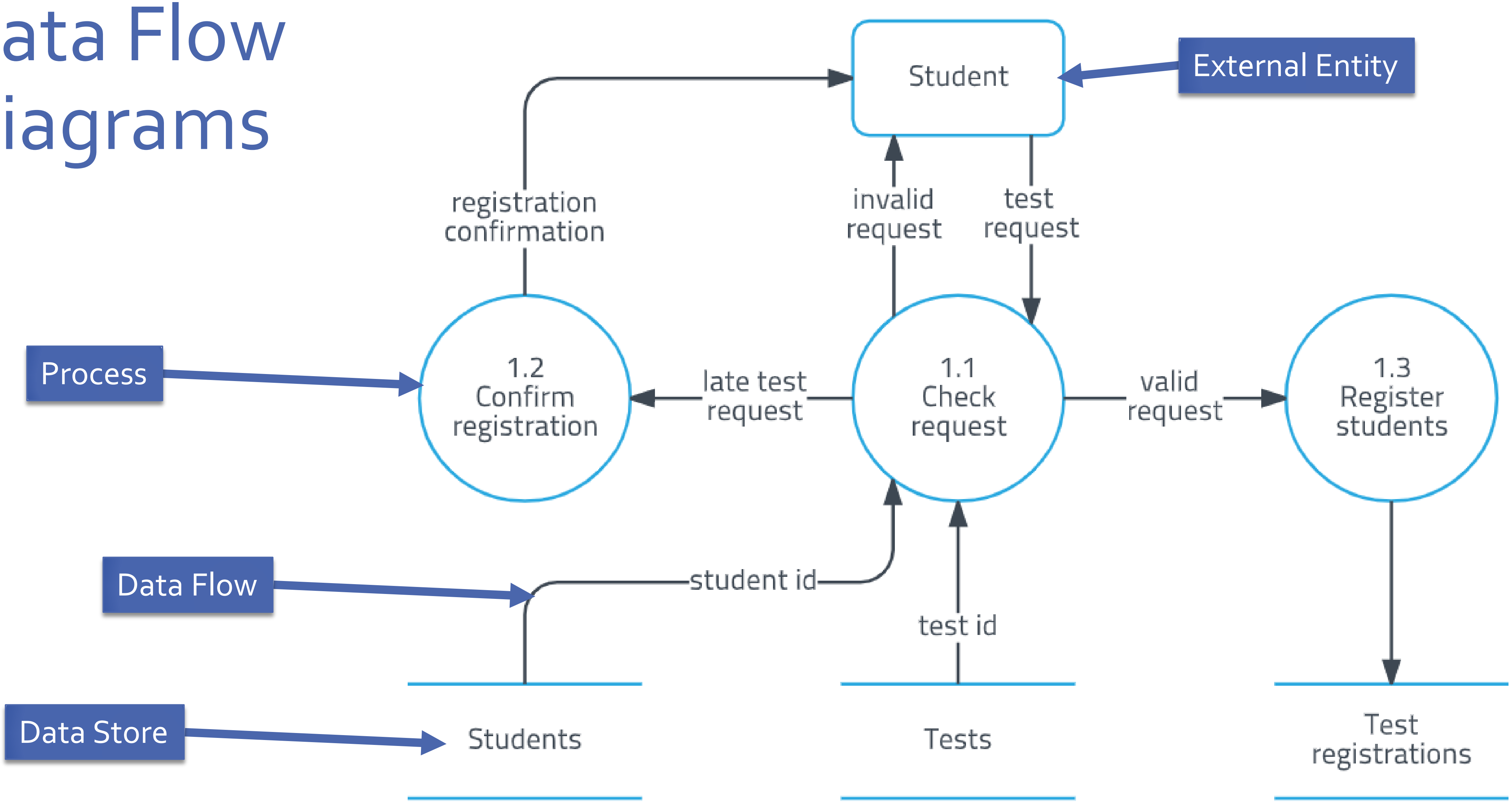
Data Flow Diagrams

Inputs of the product

Outputs of the product

Description of the behaviour

Data Flow Diagrams



Source: https://www.lucidchart.com/pages/templates/data-flow-diagram/simple-data-flow-diagram-template

Data Flow Diagrams Principles

- A process cannot create data
- Data must come from an identified source
- Data cannot move or change by itself
- A diagram must begin and end with a data store or an external system
- Processes are actions, not entities

Source: W. Stevens, G. Myers, L. Constantine, "Structured Design", IBM Systems Journal, 13 (2), 115-139, 1974.

Non-Functional Requirements

= Quality Requirements

- Accuracy
- Dependability
- Security
- Usability
- Efficiency
- Performance
- Maintainbility

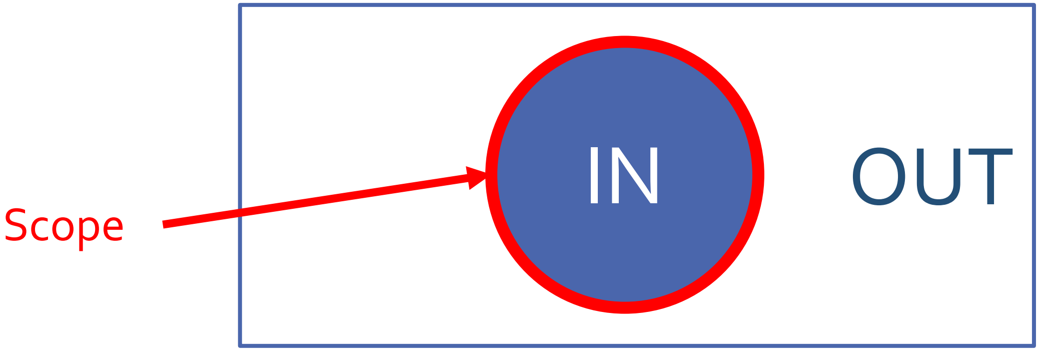
Product Vision vs Scope

Vision

- Value of a product to the client
- Purpose of the product
- Needs the projects solves
- Changes to the project should NOT change the vision

Scope

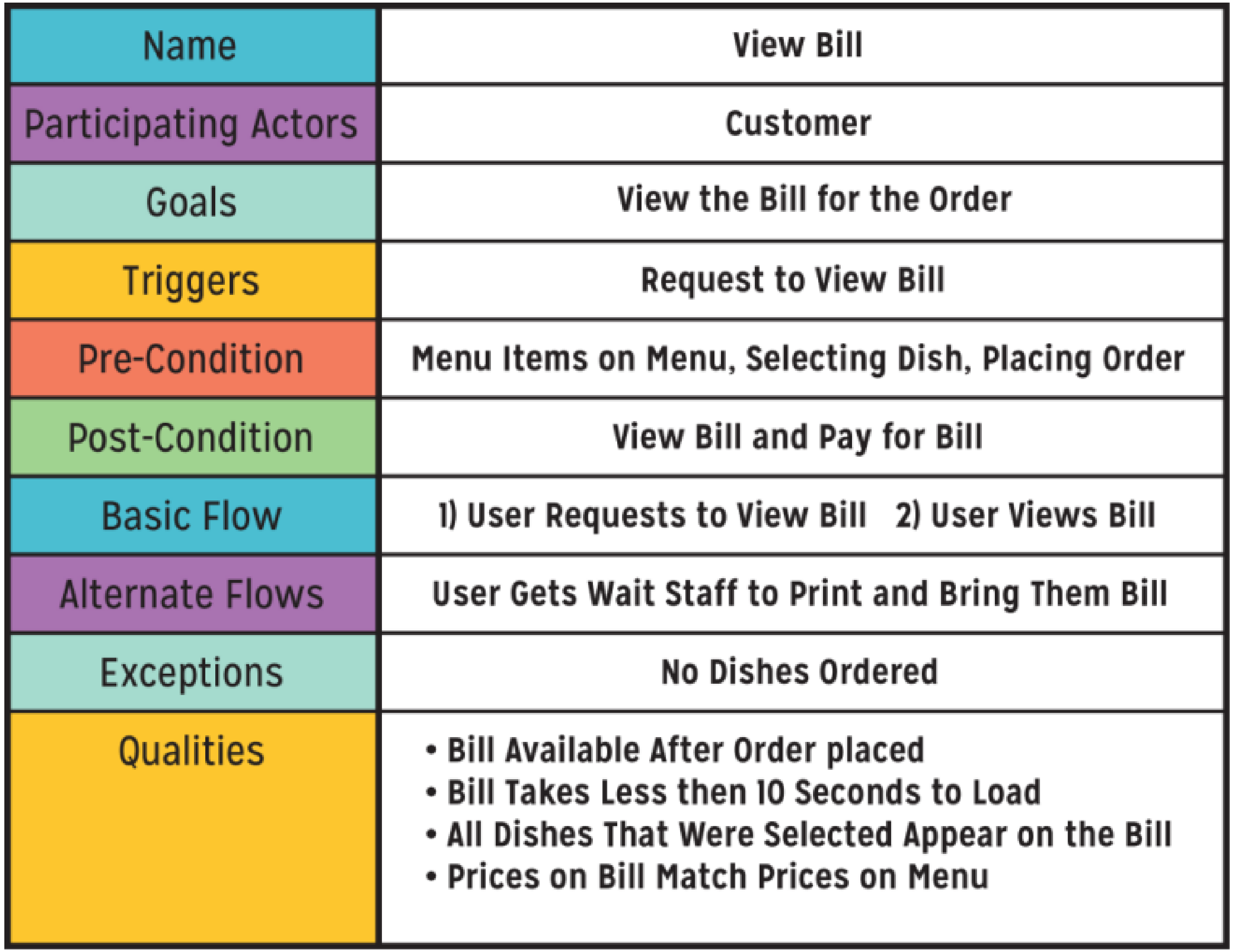
- What a project can realistically achieve
- "Boundary between what's in and what's out for the project" *(Wiegers, 2012, p. 1)*



How to Involve Clients and Users?

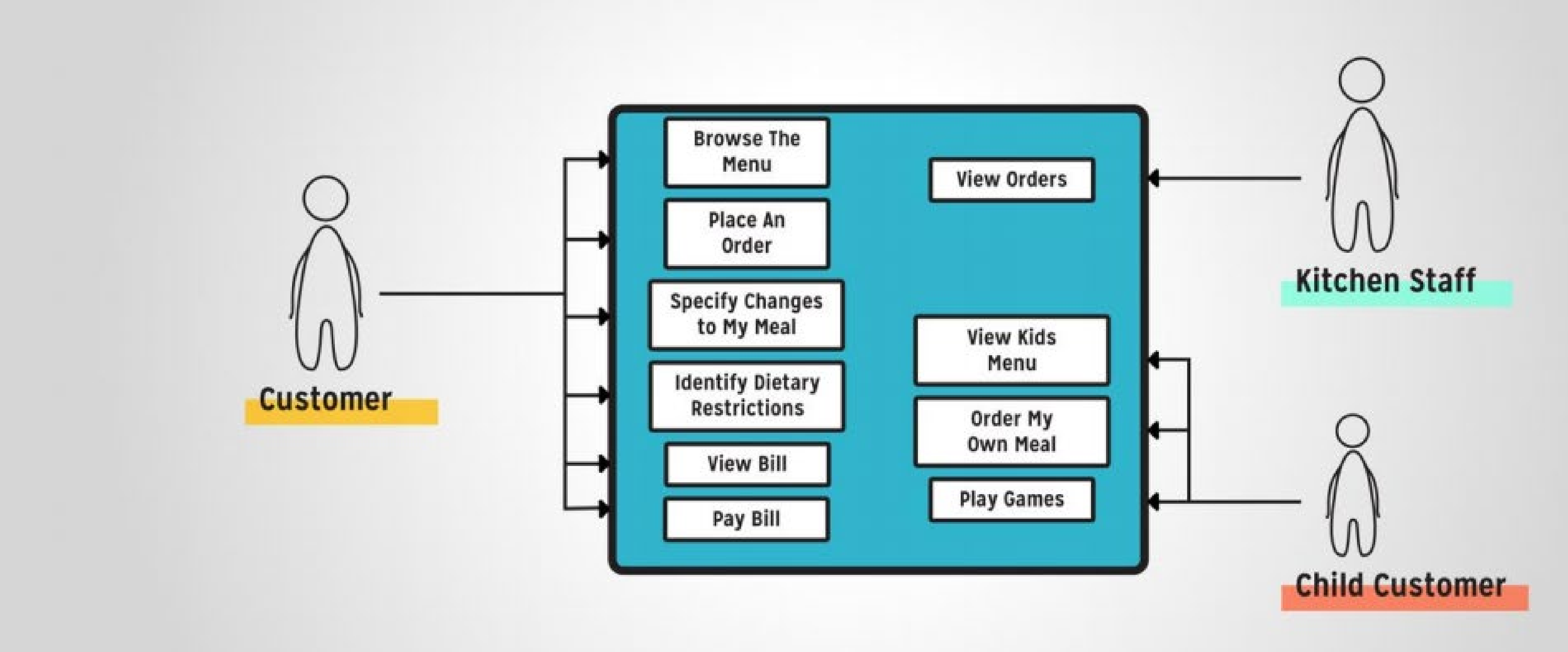
- Active collaboration with clients
- Interviews with users
- Feasibility studies with focus groups
- Observing how users use the product
- Studying previous products

Use Case Table



Name	View Bill
Participating Actors	Customer
Goals	View the Bill for the Order
Triggers	Request to View Bill
Pre-Condition	Menu Items on Menu, Selecting Dish, Placing Order
Post-Condition	View Bill and Pay for Bill
Basic Flow	1) User Requests to View Bill 2) User Views Bill
Alternate Flows	User Gets Wait Staff to Print and Bring Them Bill
Exceptions	No Dishes Ordered
Qualities	• Bill Available After Order placed • Bill Takes Less then 10 Seconds to Load • All Dishes That Were Selected Appear on the Bill • Prices on Bill Match Prices on Menu

Use Case Diagram



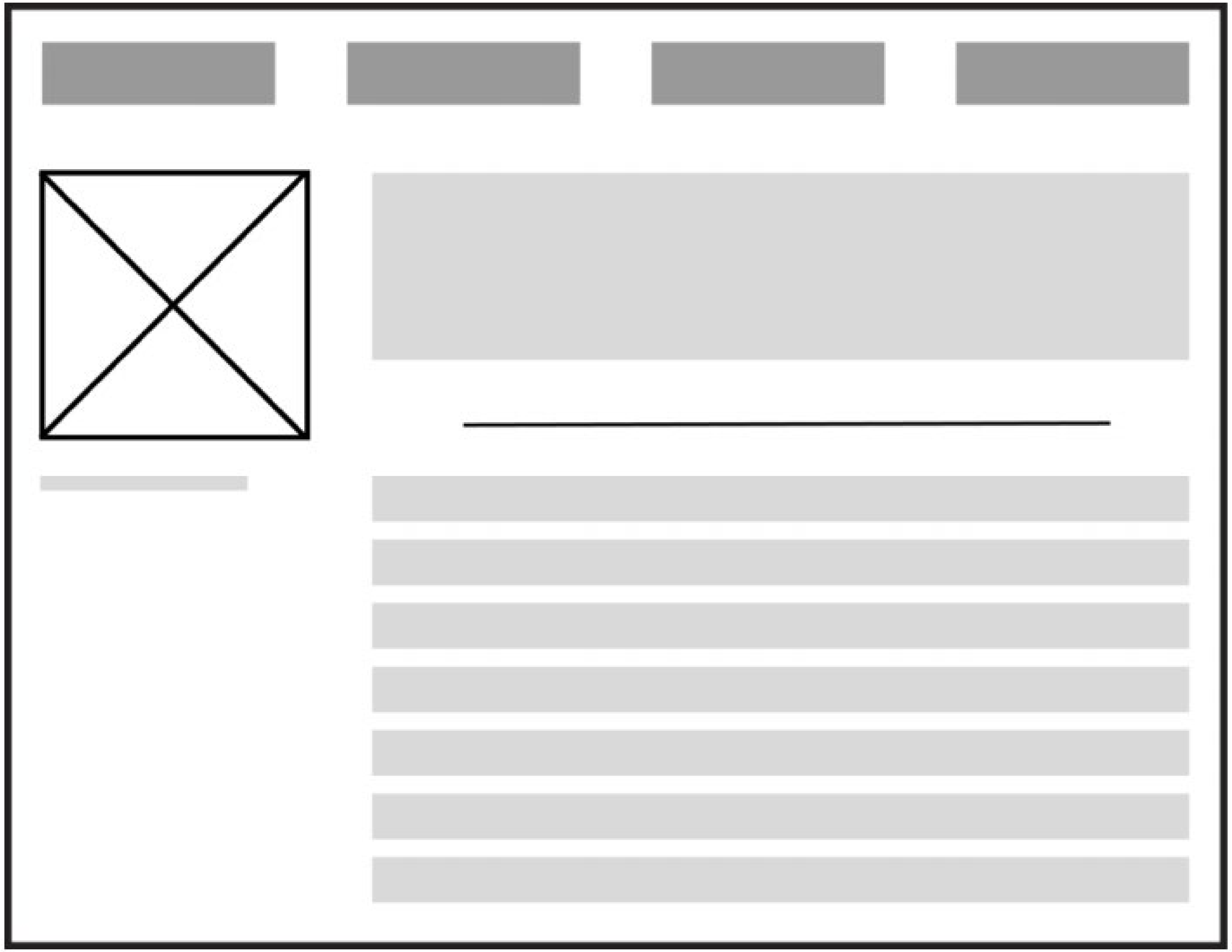
Wireframes

= Mock-ups

- Getting an idea for what will be developed
- Demonstrating ideas to clients or users; encouraging their feedback
- Communication among the development team
- Helping the client or users communicate with the product manager and team

Wireframes

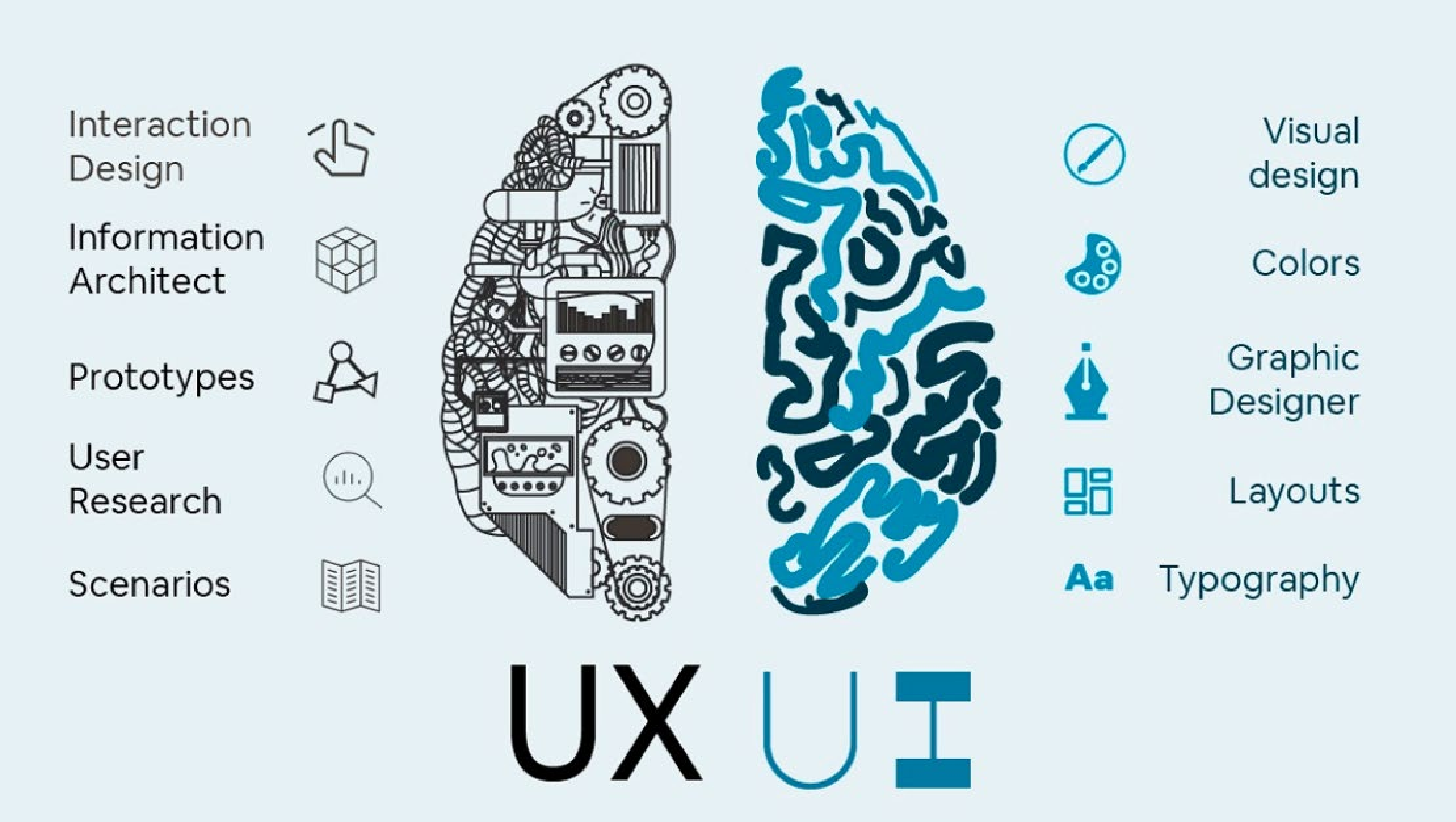
- ✓ Buttons
- ✓ Placeholders
- ✓ Text fields
- ✓ Blocks
- ✗ Colors
- ✗ Images



Some Recommended Tools

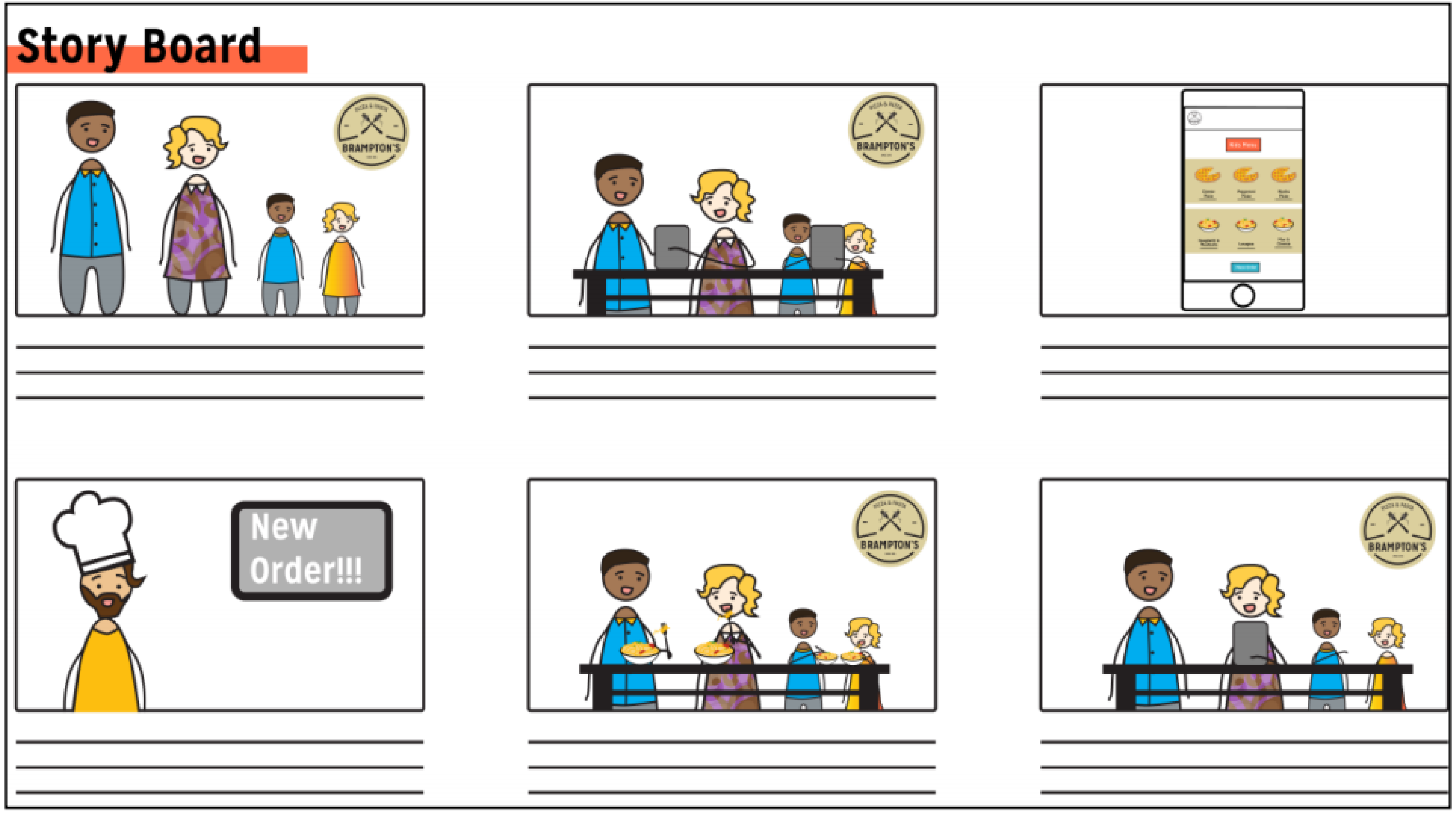
- Pen and paper!
- **Balsamiq** – **we have a license (see Slack/eClass)!**
- Figma
- Cacoo
- Mockflow
- Wireframe.cc
- Adobe Xd
- Microsoft Visio

UX vs UI

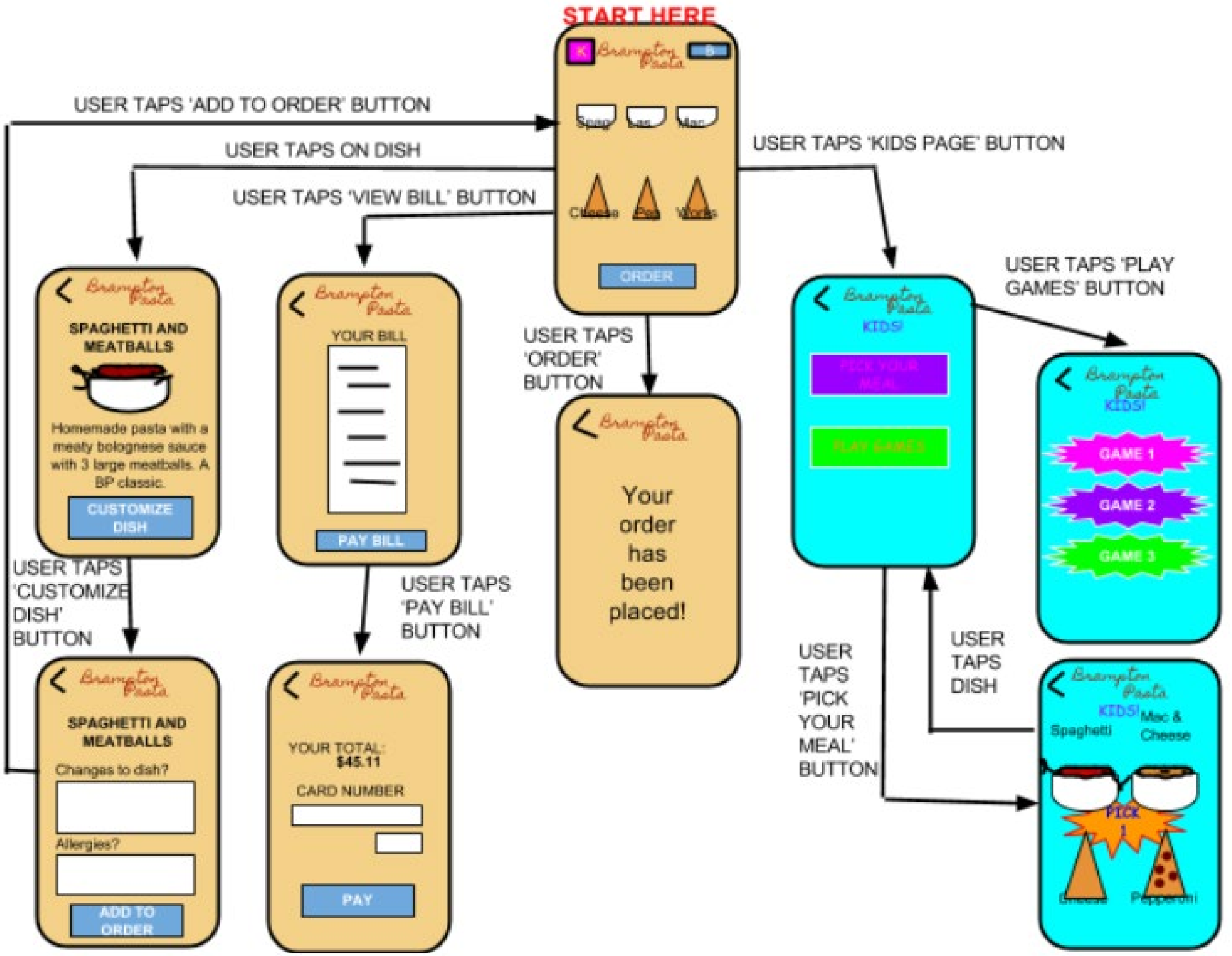


Source: https://www.studylinkclasses.com/ui-ux.php

UX Storyboards



UI Storyboards



START HERE

USER TAPS 'ADD TO ORDER' BUTTON

USER TAPS ON DISH

USER TAPS 'VIEW BILL' BUTTON

USER TAPS 'KIDS PAGE' BUTTON

USER TAPS 'ORDER' BUTTON

USER TAPS 'PLAY GAMES' BUTTON

USER TAPS 'CUSTOMIZE DISH' BUTTON

USER TAPS 'PAY BILL' BUTTON

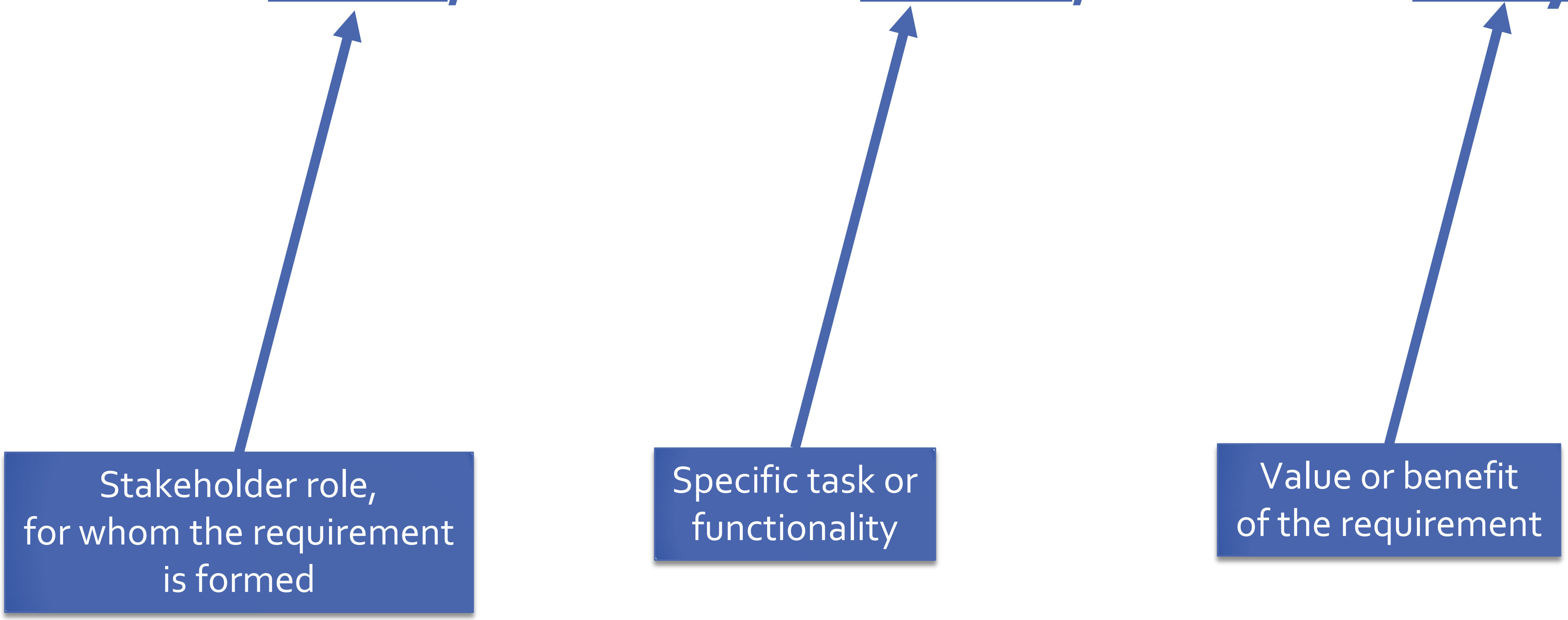
USER TAPS 'PICK YOUR MEAL' BUTTON

USER TAPS DISH

User Stories

Consistent format!

“As a **who**, I want to **what**, so that **why**.”



Stakeholder role, for whom the requirement is formed

Specific task or functionality

Value or benefit of the requirement

User Story Example

Who **What**

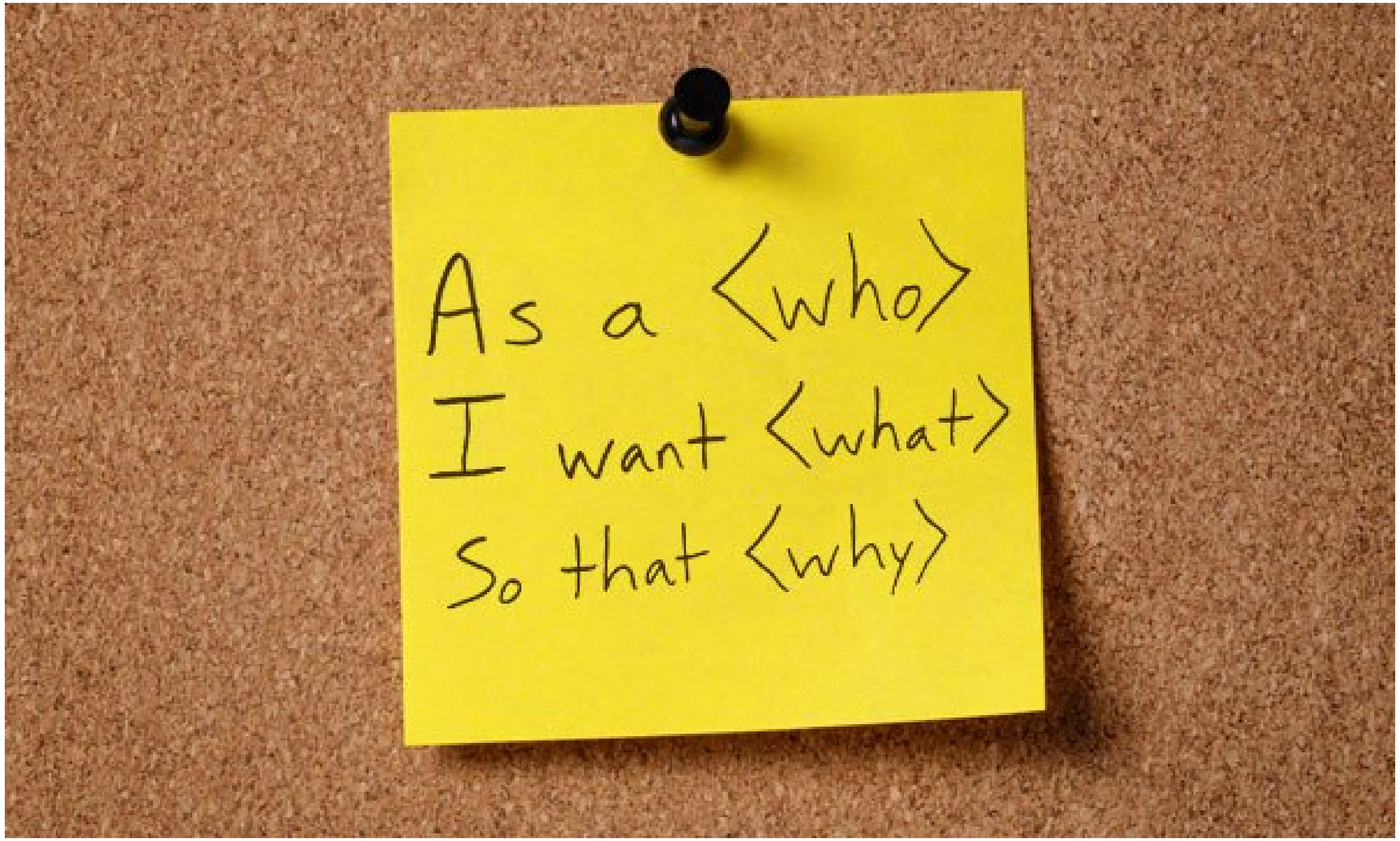
As a customer, I want to be able to identify

Why

dietary restrictions, so that I know I can eat

the food I order.

User Story Cards / Sticky Notes



As a <who>
I want <what>
So that <why>

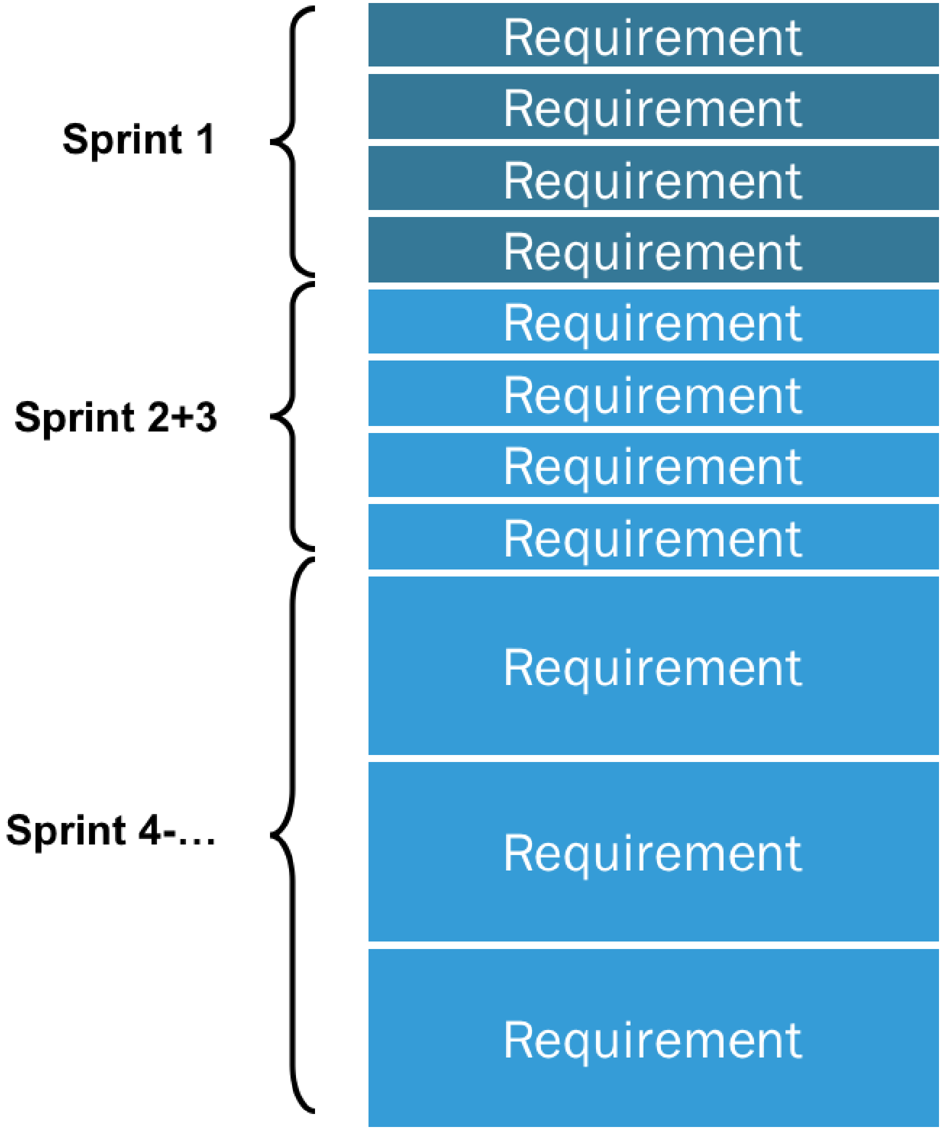
Source: https://www.ae.be/blog-en/top-tools-user-story-mapping-post-its-best-digital-apps/

Avoid “Epic User Stories”

- “Epic user story” contains descriptions that are too vague or broad
- Can be difficult to estimate how long it will take
- **Best strategy:** Provide just enough information for a developer to understand how to implement it, but not so much information that implementation details become part of the story

Product Backlog

- **List of software features, sorted by priority**
- Helps to organize work, prioritize tasks
- Critical to Scrum / Agile

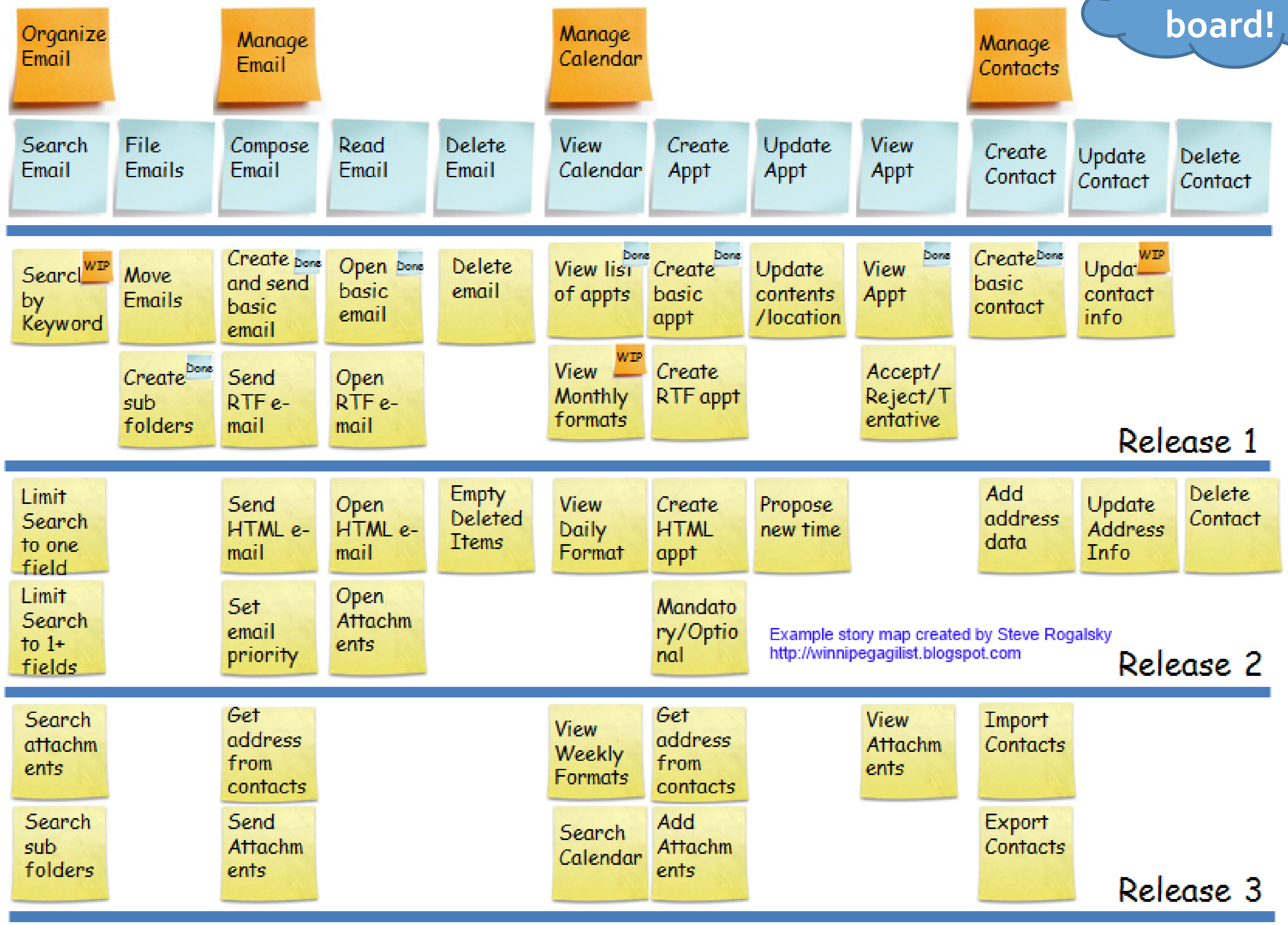


Source: https://www.scrum.org/resources/what-is-a-product-backlog

Story Map

- Visualized backlog!
- **Each card = User Story**
- Column = Category
- Within each column, stories are sorted by priority
- Excellent example of the Agile principle
- **You're highly encouraged to use story maps!**

NOT a Kanban board!



	Organize Email		Manage Email			Manage Calendar				Manage Contacts		
	Search Email	File Emails	Compose Email	Read Email	Delete Email	View Calendar	Create Appt	Update Appt	View Appt	Create Contact	Update Contact	Delete Contact
Release 1	Search by Keyword WIP	Move Emails	Create and send basic email Done	Open basic email Done	Delete email	View list of appts Done	Create basic appt Done	Update contents /location	View Appt Done	Create basic contact Done	Update contact info WIP	
		Create sub folders Done	Send RTF e-mail	Open RTF e-mail		View Monthly formats WIP	Create RTF appt		Accept/Reject/Tentative			
Release 2	Limit Search to one field		Send HTML e-mail	Open HTML e-mail	Empty Deleted Items	View Daily Format	Create HTML appt	Propose new time		Add address data	Update Address Info	Delete Contact
	Limit Search to 1+ fields		Set email priority	Open Attachments			Mandatory/Optional					
Release 3	Search attachments		Get address from contacts			View Weekly Formats	Get address from contacts		View Attachments	Import Contacts		
	Search sub folders		Send Attachments			Search Calendar	Add Attachments			Export Contacts		

Example story map created by Steve Rogalsky
http://winnipegagilist.blogspot.com

Source: *http://winnipegagilist.blogspot.in/2012/03/how-to-create-user-story-map.html*

Criteria for User Stories: INVEST model

Independent

Negotiable

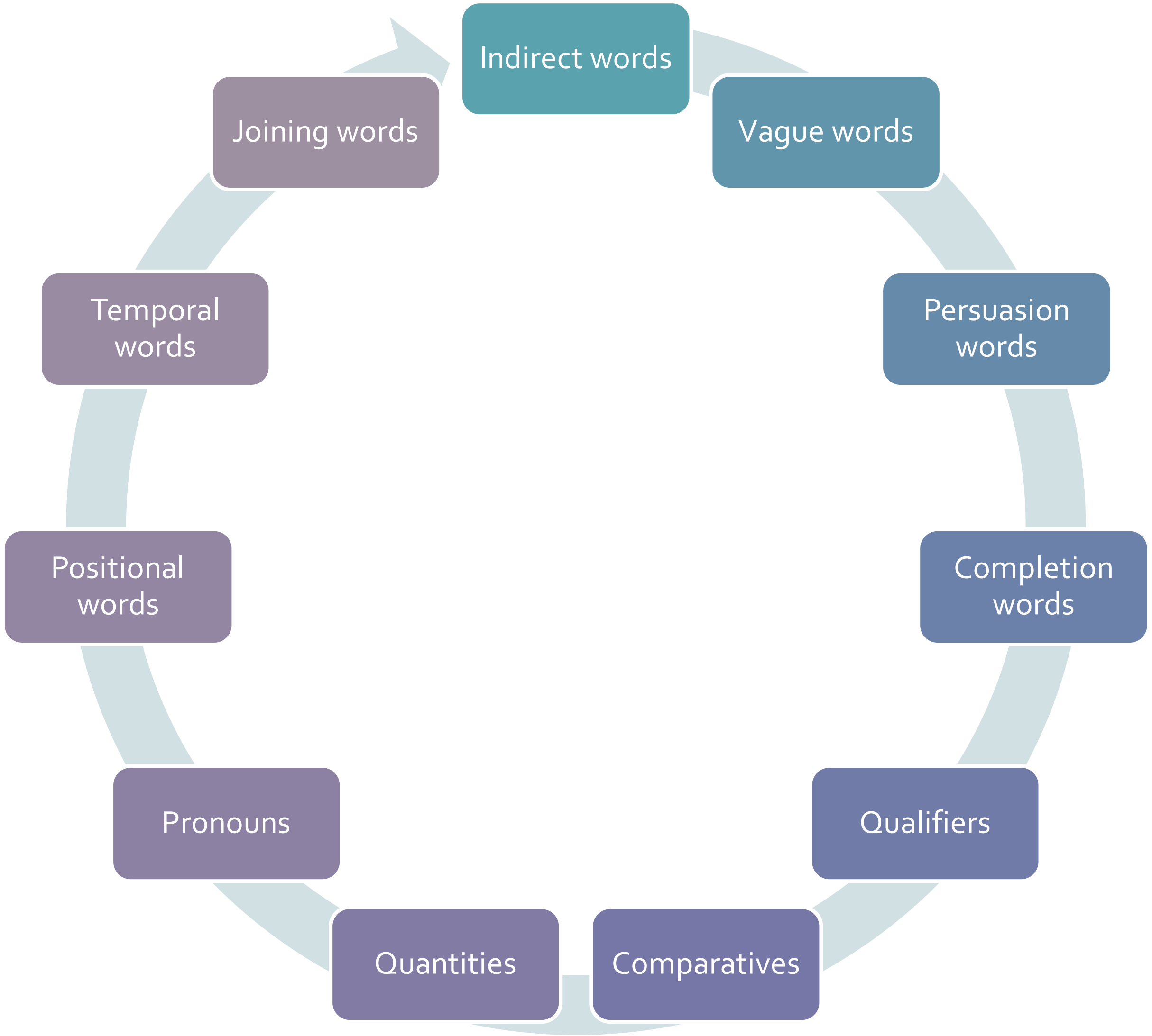
Valuable

Estimatable

Small

Testable

Ambiguous Requirements



Indirect words

Vague words

Persuasion words

Completion words

Qualifiers

Comparatives

Quantities

Pronouns

Positional words

Temporal words

Joining words

QUESTIONS?!