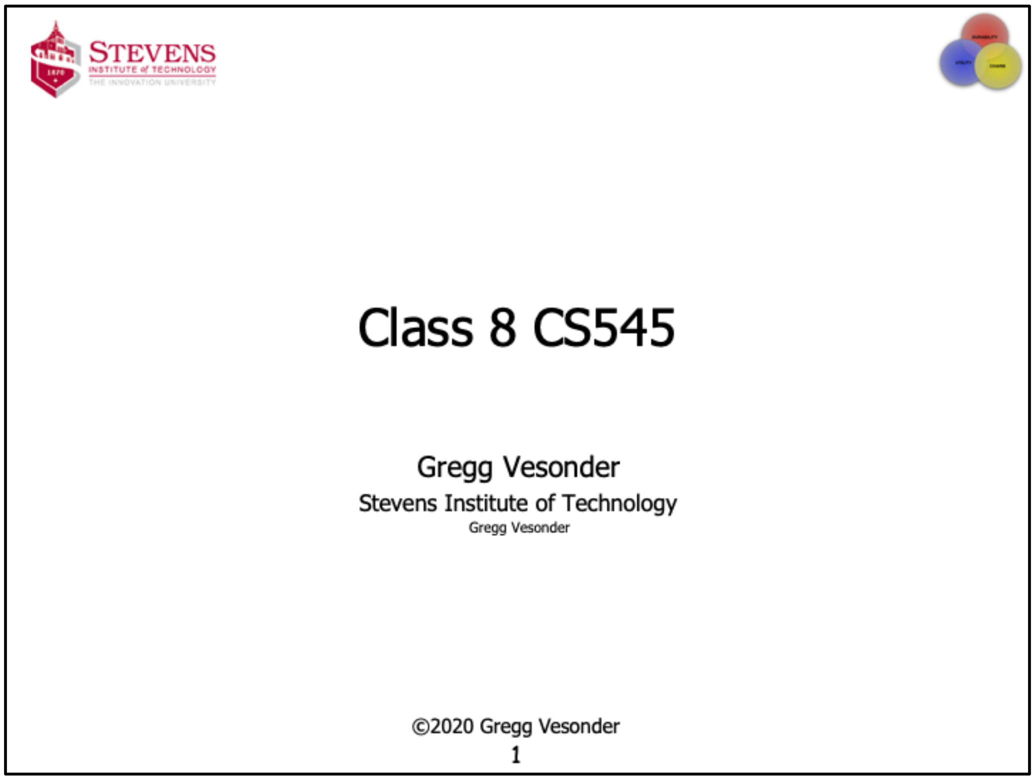
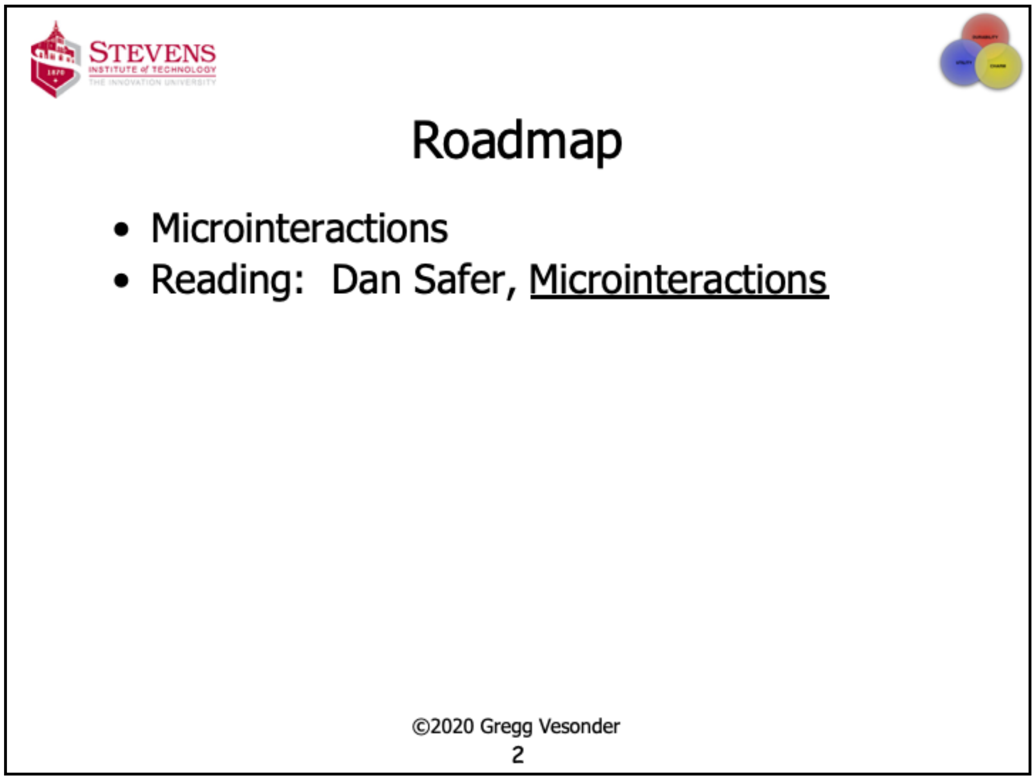


Class 8 CS545

Gregg Vesonder
Stevens Institute of Technology
Gregg Vesonder

1

Welcome to class 8.



Roadmap

- Microinteractions
- Reading: Dan Safer, Microinteractions

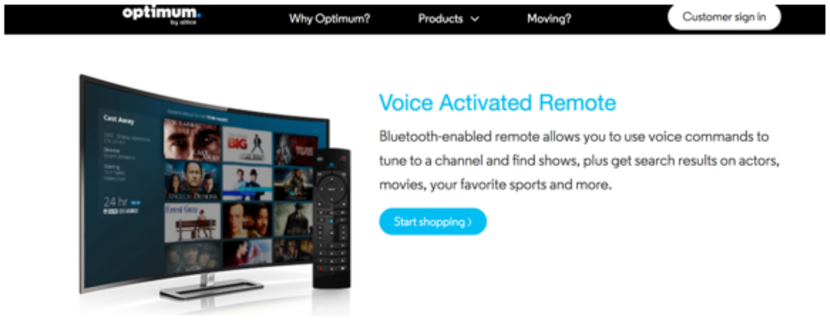
©2020 Gregg Vesonder

2

This lecture is all about microinteractions and covers the first section of the book. You need not get the book to understand it. After discussing the project with some of the teams I think understanding and focusing on microinteractions will be helpful. We will also cover more usability patterns in later lectures.

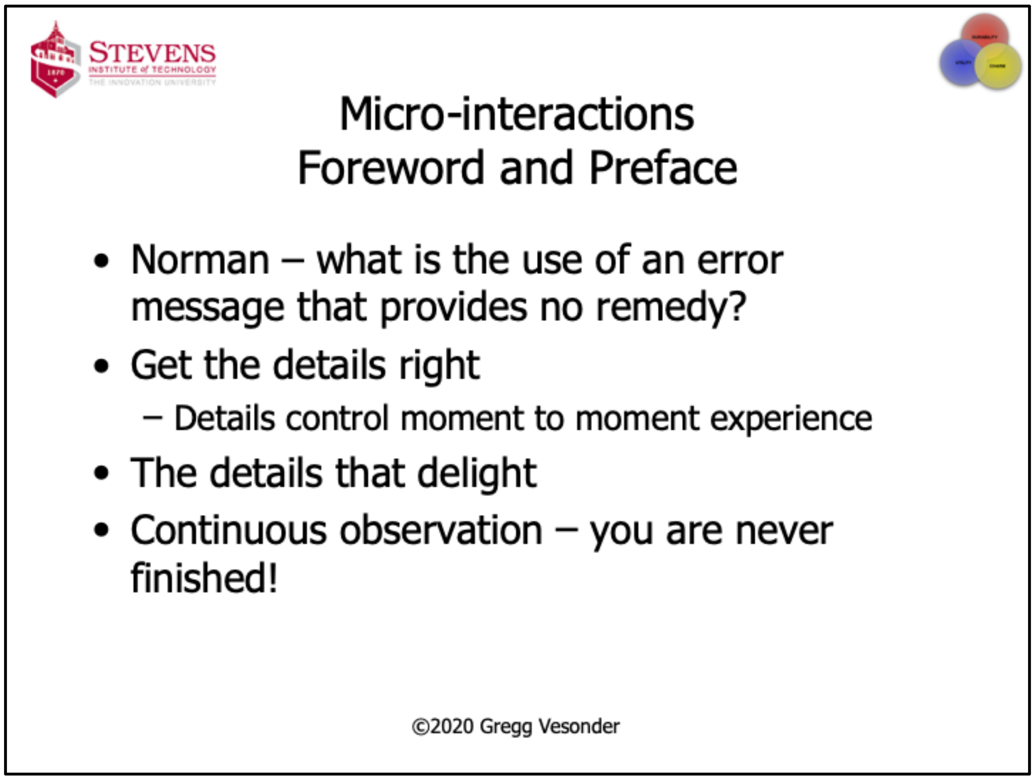
LogBook

- Is speech the answer to the multiple tv remote control issue?



Voice Activated Remote

Bluetooth-enabled remote allows you to use voice commands to tune to a channel and find shows, plus get search results on actors, movies, your favorite sports and more.

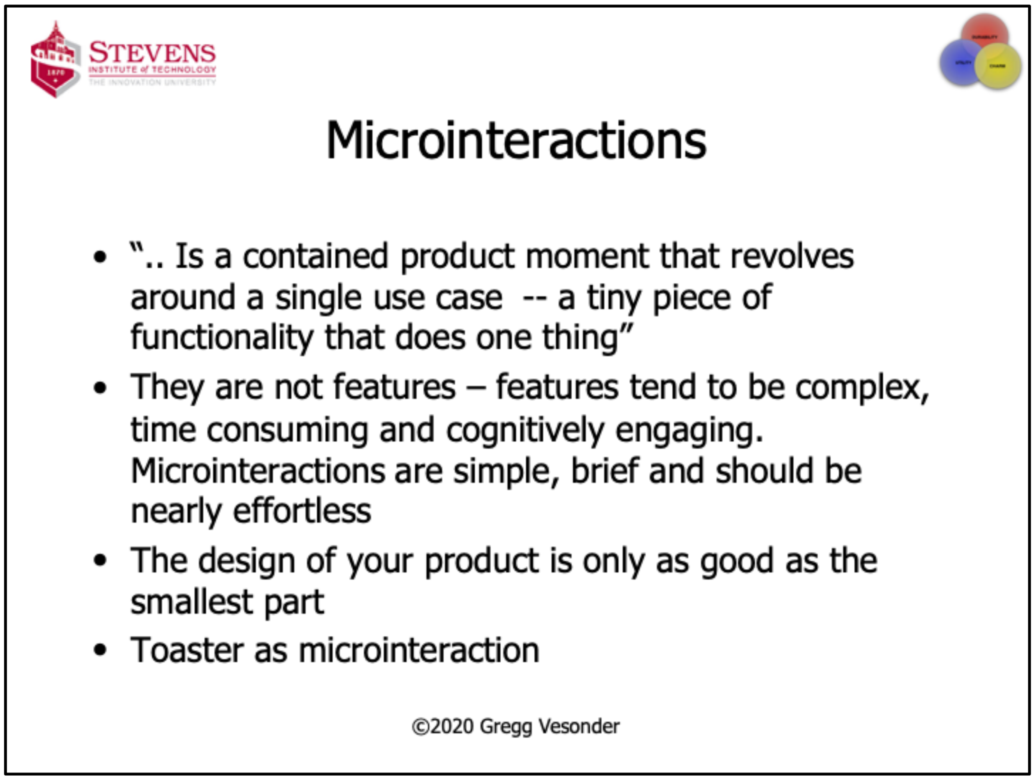


Micro-interactions Foreword and Preface

- Norman – what is the use of an error message that provides no remedy?
- Get the details right
  - Details control moment to moment experience
- The details that delight
- Continuous observation – you are never finished!

Most of the slides in this lecture are self explanatory, especially if you have read the book. I think that this is the precise time for you to cover microinteractions, after actually trying to build a system that focuses on User Centered Design. The value of making the complex as simple as possible becomes much more clear once you try to build something that you want folks to use. The difference in an "amateur" app and one that is successful is focusing on the details and ensuring that the users have a delightful and productive user experience that is not frustrating or annoying.

Norman provides an example of one of my aggravations – an error message that does not help you to correct or at least learn from the error. I bet you encounter at least one of those error messages every day.

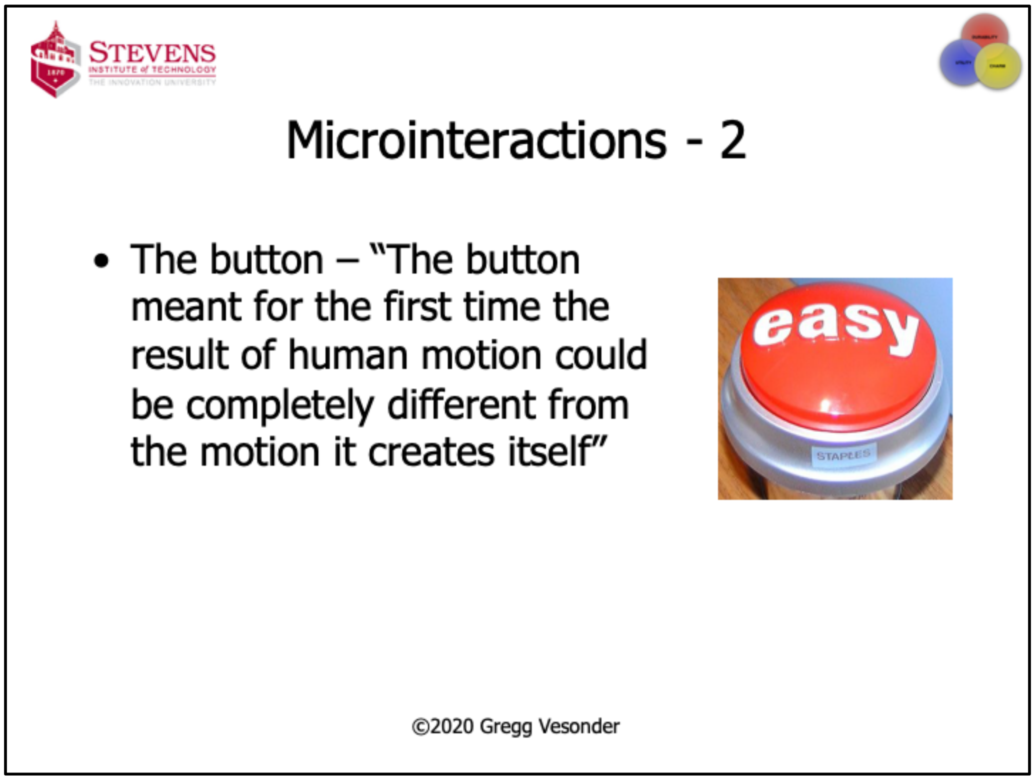


STEVENS INSTITUTE of TECHNOLOGY

Microinteractions

- ".. Is a contained product moment that revolves around a single use case -- a tiny piece of functionality that does one thing"
- They are not features – features tend to be complex, time consuming and cognitively engaging. Microinteractions are simple, brief and should be nearly effortless
- The design of your product is only as good as the smallest part
- Toaster as microinteraction

Microinteractions are small and usually a collection of them result in a feature – that is if you are careful enough to design the feature as a series of interactions! A toaster is a microinteraction – place the bread in the machine and out pops toast – if you plugged in the toaster!

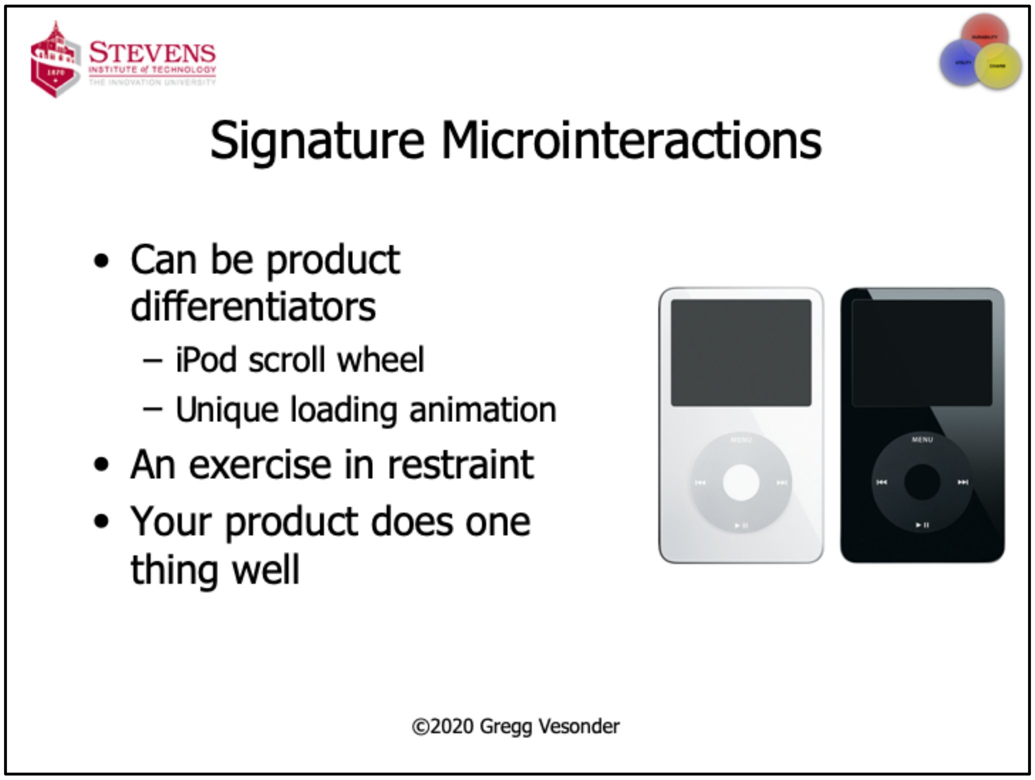


Microinteractions - 2

- The button – "The button meant for the first time the result of human motion could be completely different from the motion it creates itself"

©2020 Gregg Vesonder

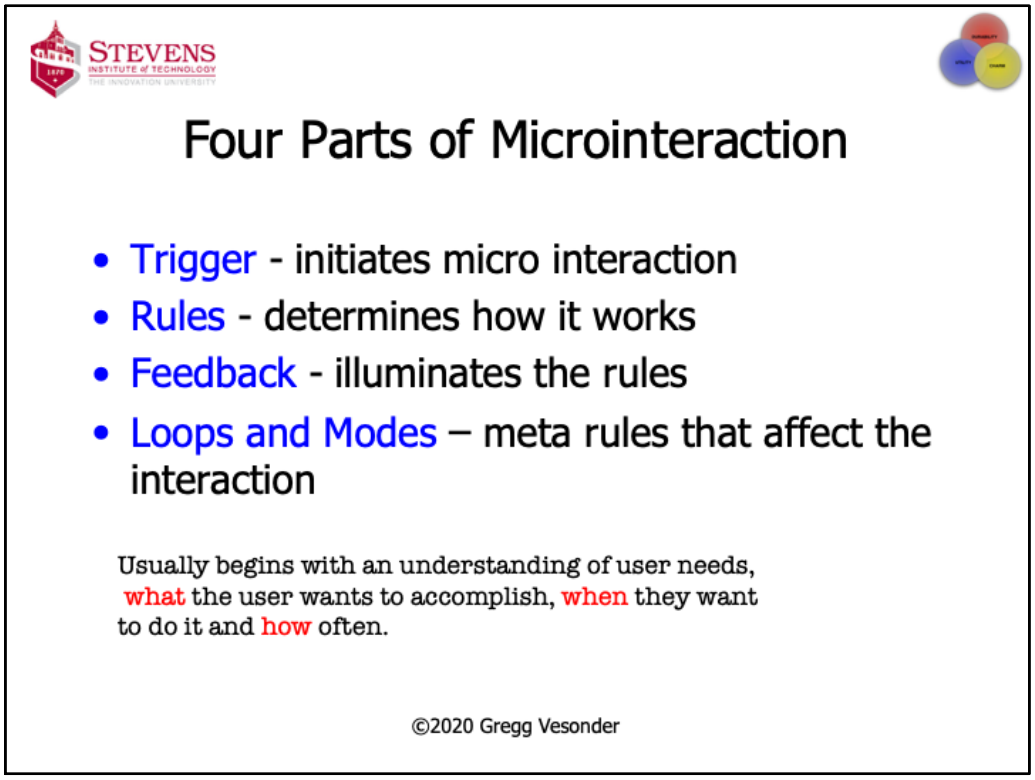
Buttons were the first widely developed microinteraction. A simple press could result in significant activity, such as starting an engine. Buttons signify masking complexity as was clearly evident from Staples ad campaign that emphasized how easy it was to do order supplies form them. The microinteraction of the "easy" button became the corporate signature.



Signature Microinteractions

- Can be product differentiators
  - iPod scroll wheel
  - Unique loading animation
- An exercise in restraint
- Your product does one thing well

Just like the Metaphorical example of the Staples easy button, microinteractions can provide a product signature – something done well to make an interaction simpler often becomes a prime identifier of the product such as the simplicity of the iPod controls.

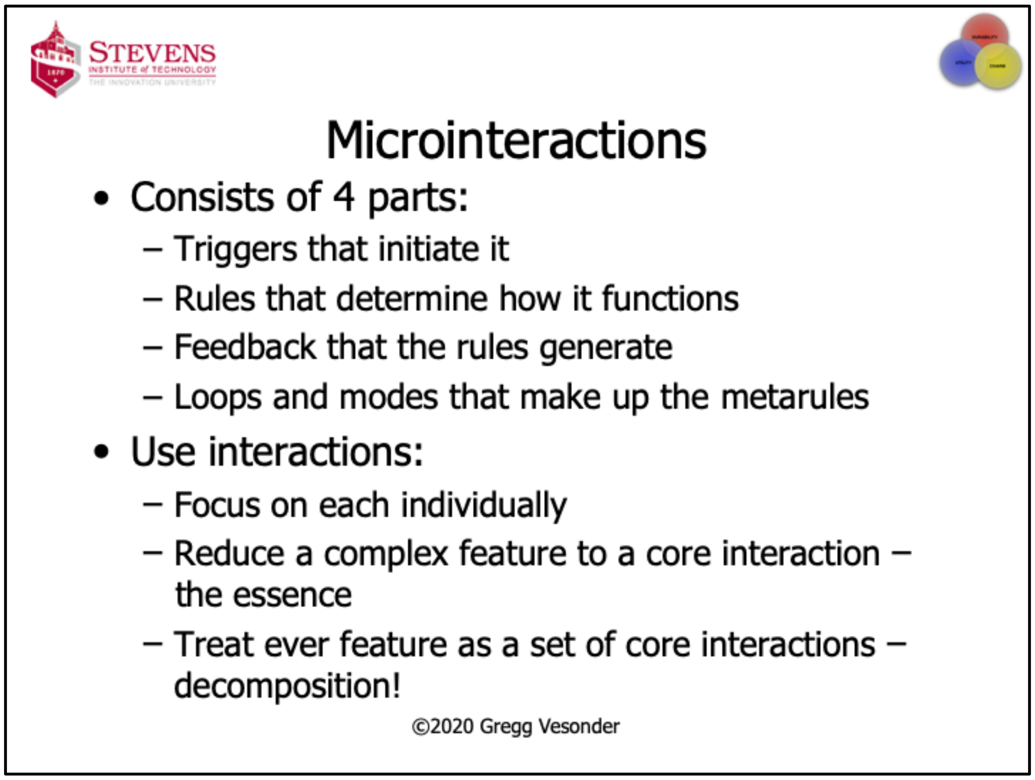


Four Parts of Microinteraction

- Trigger - initiates micro interaction
- Rules - determines how it works
- Feedback - illuminates the rules
- Loops and Modes – meta rules that affect the interaction

Usually begins with an understanding of user needs, what the user wants to accomplish, when they want to do it and how often.

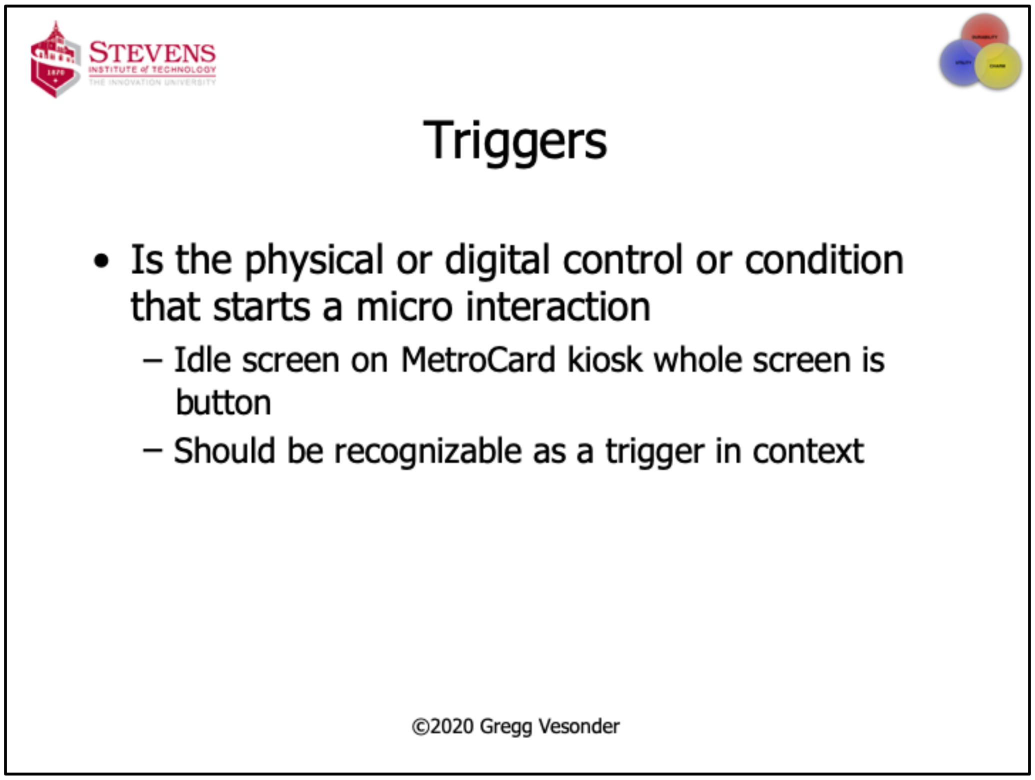
So let’s go a bit more deeply into triggers



Microinteractions

- Consists of 4 parts:
  - Triggers that initiate it
  - Rules that determine how it functions
  - Feedback that the rules generate
  - Loops and modes that make up the metarules
- Use interactions:
  - Focus on each individually
  - Reduce a complex feature to a core interaction – the essence
  - Treat ever feature as a set of core interactions – decomposition!

So the key is to take a feature and decompose it into the set of core interactions – eliminating as much as possible in order to simplify.

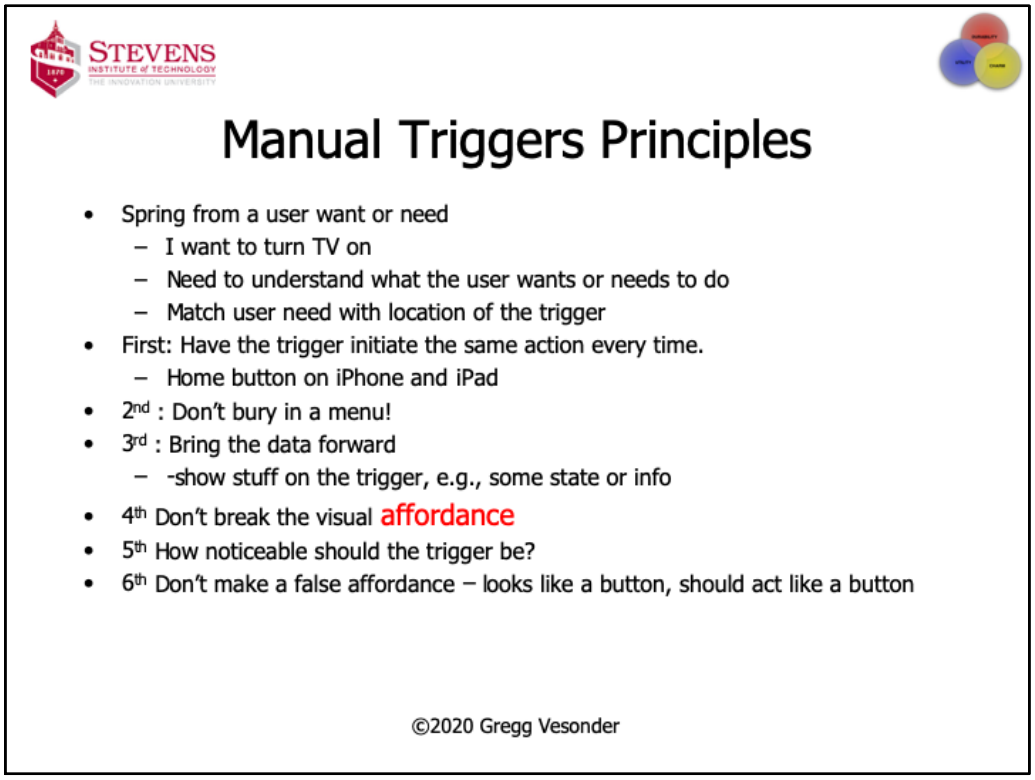


Triggers

- Is the physical or digital control or condition that starts a micro interaction
  - Idle screen on MetroCard kiosk whole screen is button
  - Should be recognizable as a trigger in context

©2020 Gregg Vesonder

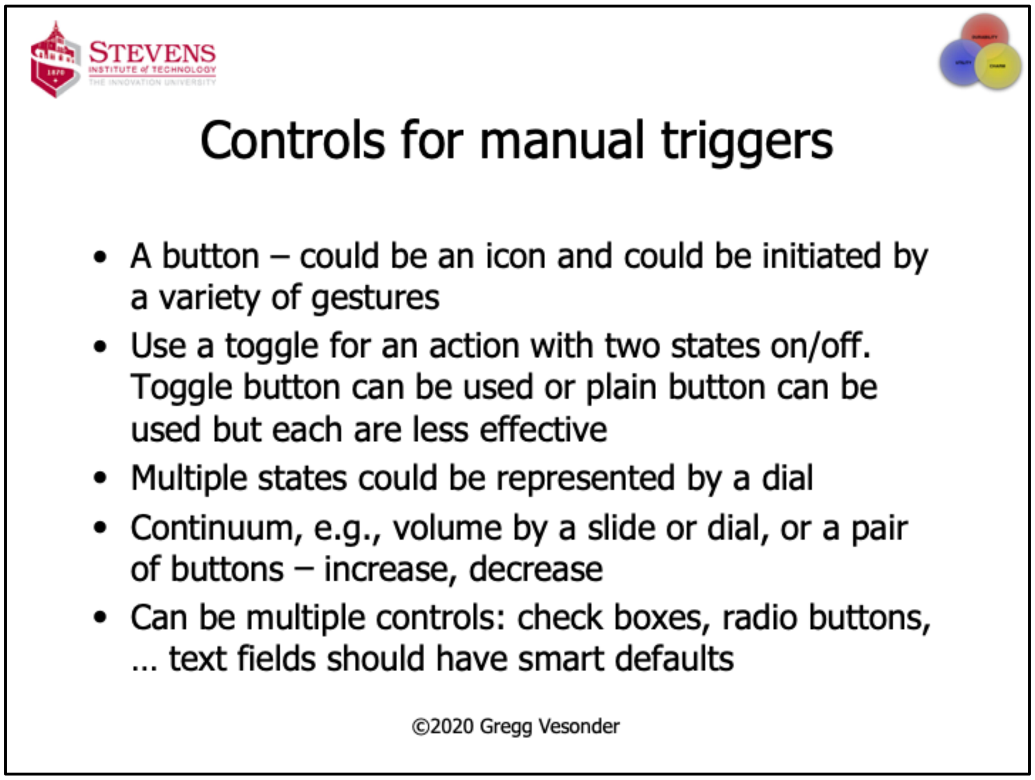
So a trigger starts the sequence – Safer has a wonderful description of the MTA's screen which invites you to touch it – this is the trigger for the interaction of buying a ticket the invitation to touch the screen. Ingeniously they designed the interaction so that each screen requires only one action. Think about that next time you are buying a ticket for the MTA or NJ Transit.



Manual Triggers Principles

- Spring from a user want or need
  - I want to turn TV on
  - Need to understand what the user wants or needs to do
  - Match user need with location of the trigger
- First: Have the trigger initiate the same action every time.
  - Home button on iPhone and iPad
- 2nd : Don't bury in a menu!
- 3rd : Bring the data forward
  - -show stuff on the trigger, e.g., some state or info
- 4th Don't break the visual affordance
- 5th How noticeable should the trigger be?
- 6th Don't make a false affordance – looks like a button, should act like a button

A user need is to buy a ticket, turn on the tv, … . It is best to have the trigger initiate the same action every time but in this world of minimalist design of hardware and software, we often have the button support multiple duties – but in different contexts so this does differentiate the trigger for the user. Triggers should be highly visible – they should scream to you to interact with them!
Bringing the data forward means to represent things on the trigger, for example number of unread emails on the button that access emails.
We introduce the concept of affordance – important in HCI - an affordane is an object or symbol that shouts for it to be used in a certain manner. For example a button elicits pushing it. Or we can say it affords pushing. That is the key to an affordance – it is an object that affords an action – a chair affords sitting a slider affords sliding, ...

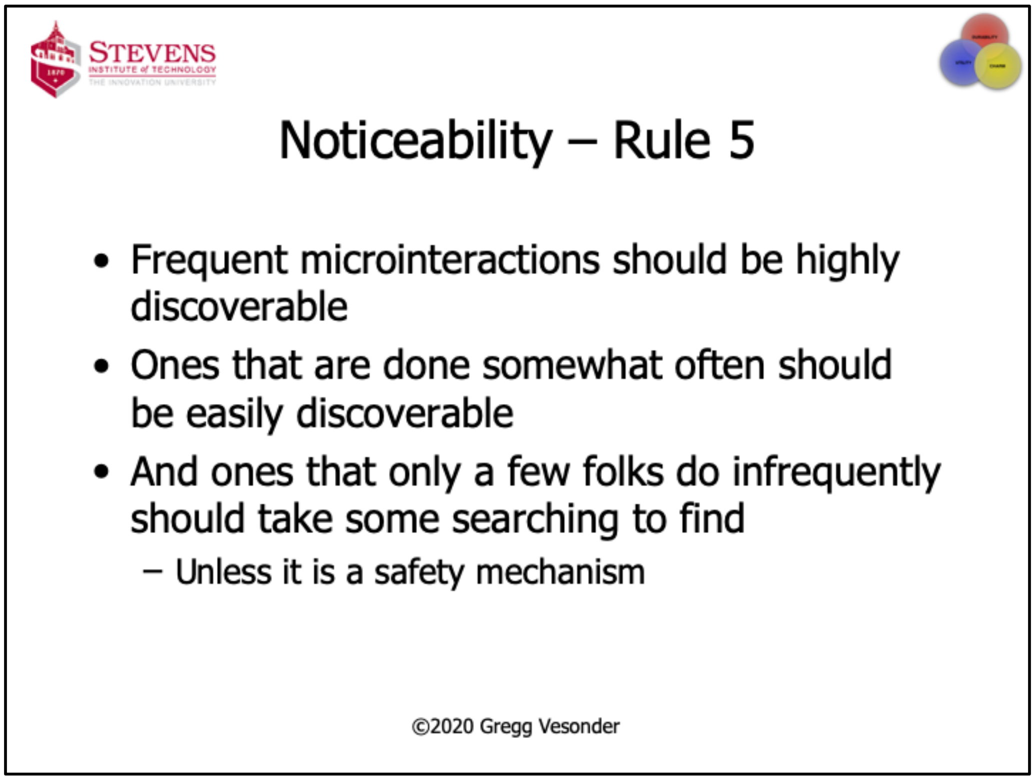


STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY

Controls for manual triggers

- A button – could be an icon and could be initiated by a variety of gestures
- Use a toggle for an action with two states on/off. Toggle button can be used or plain button can be used but each are less effective
- Multiple states could be represented by a dial
- Continuum, e.g., volume by a slide or dial, or a pair of buttons – increase, decrease
- Can be multiple controls: check boxes, radio buttons, … text fields should have smart defaults

This slide simply goes through iconic symbols for actions. Review your app to see if you have used these devices and to determine whether you have used them effectively.

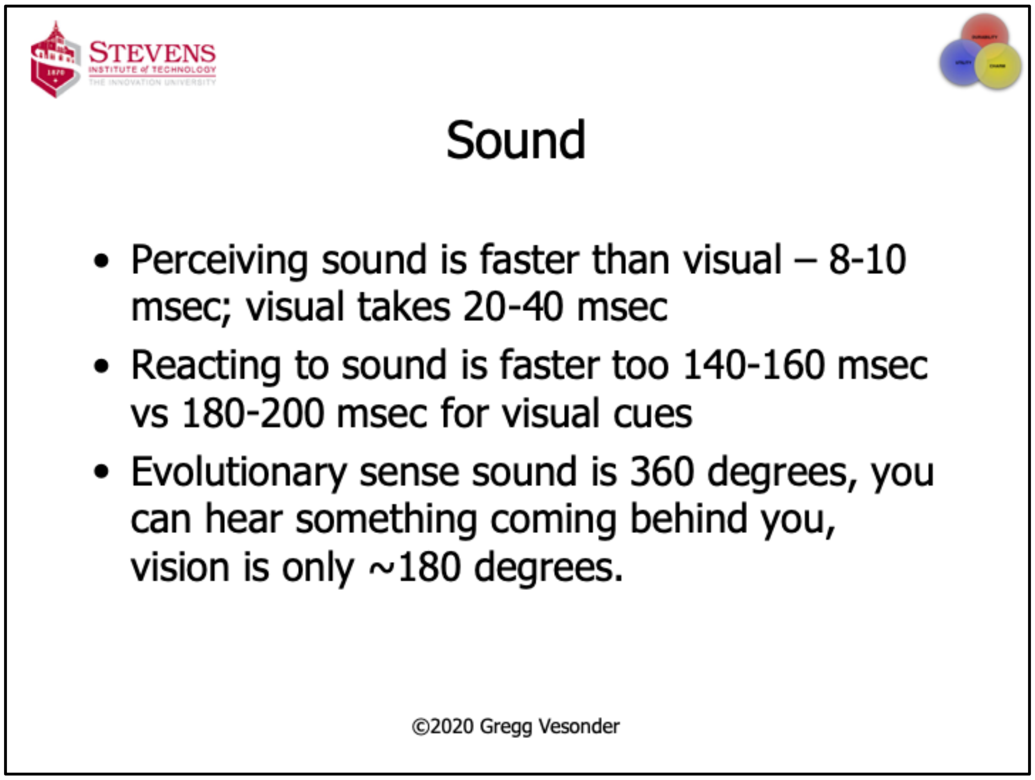


Noticeability – Rule 5

- Frequent microinteractions should be highly discoverable
- Ones that are done somewhat often should be easily discoverable
- And ones that only a few folks do infrequently should take some searching to find
  - Unless it is a safety mechanism

©2020 Gregg Vesonder

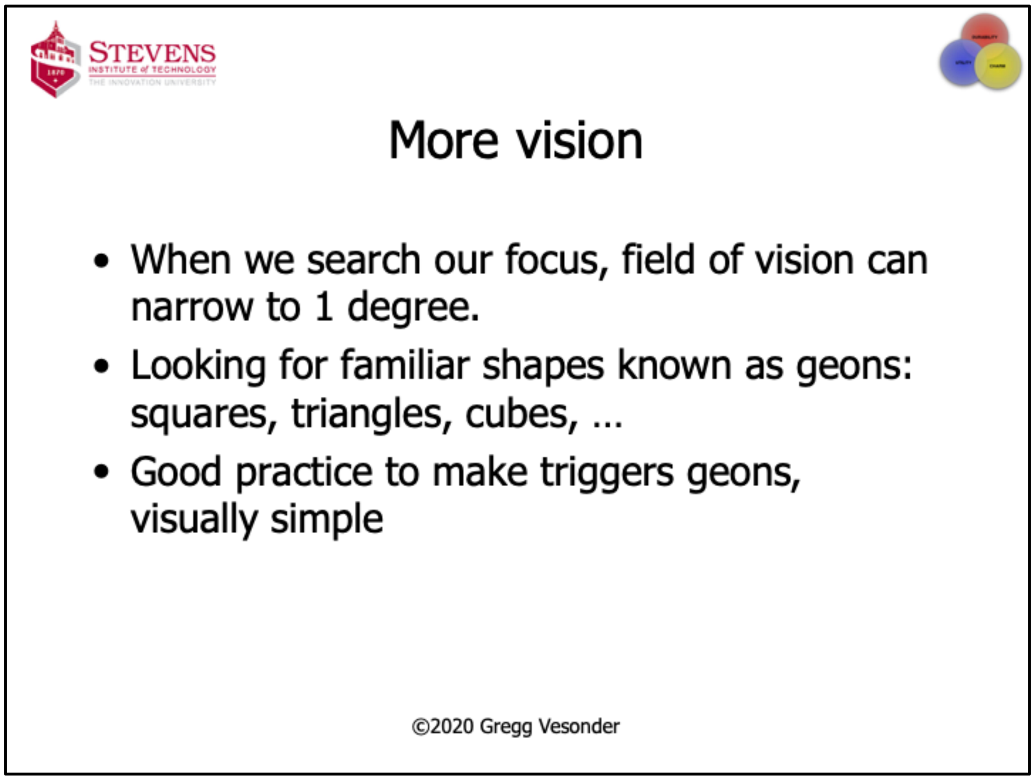
This advice is logical, with one exception. Frequent microinteractions should be very noticeable, a fancy word is salient, they leap off the screen and beckon you to push or slide. Frequent ones also should be present and infrequent interactions should require some work – except in the cast of a safety mechanism, for example a panic button on your phone.



Sound

- Perceiving sound is faster than visual – 8-10 msec; visual takes 20-40 msec
- Reacting to sound is faster too 140-160 msec vs 180-200 msec for visual cues
- Evolutionary sense sound is 360 degrees, you can hear something coming behind you, vision is only ~180 degrees.

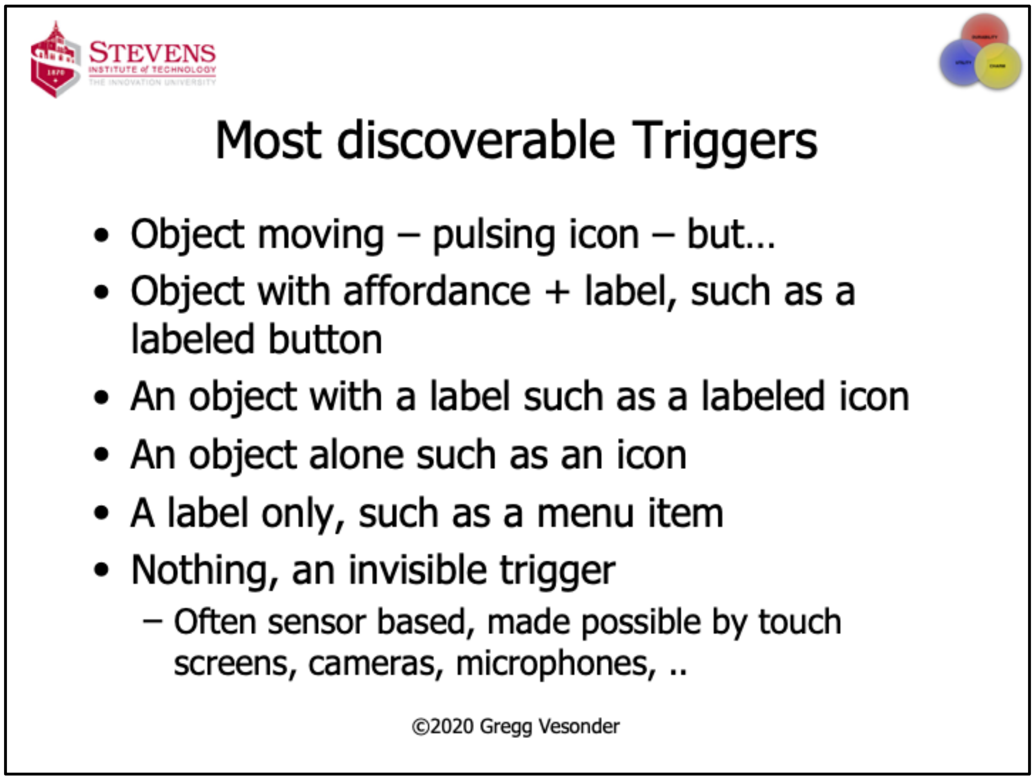
So some more psychophysics on vision and sound. Sound is a bit more effective especially for alerts



More vision

- When we search our focus, field of vision can narrow to 1 degree.
- Looking for familiar shapes known as geons: squares, triangles, cubes, …
- Good practice to make triggers geons, visually simple

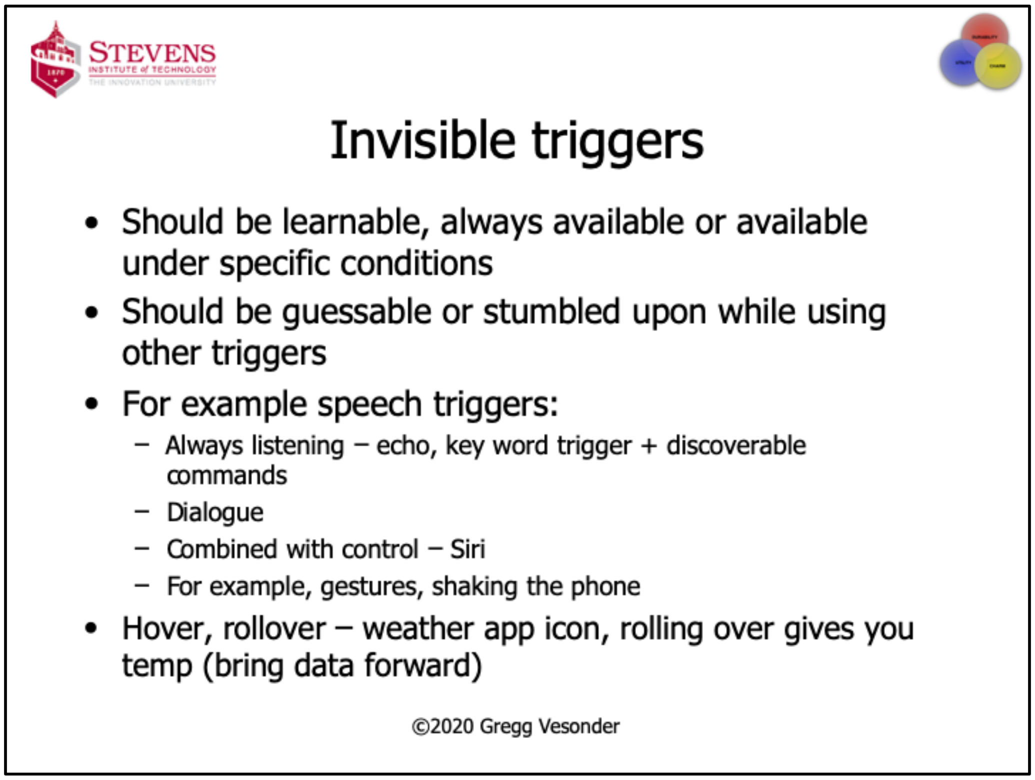
Standard shapes are easy to perceive and therefore should be used frequently



Most discoverable Triggers

- Object moving – pulsing icon – but…
- Object with affordance + label, such as a labeled button
- An object with a label such as a labeled icon
- An object alone such as an icon
- A label only, such as a menu item
- Nothing, an invisible trigger
  - Often sensor based, made possible by touch screens, cameras, microphones, ..

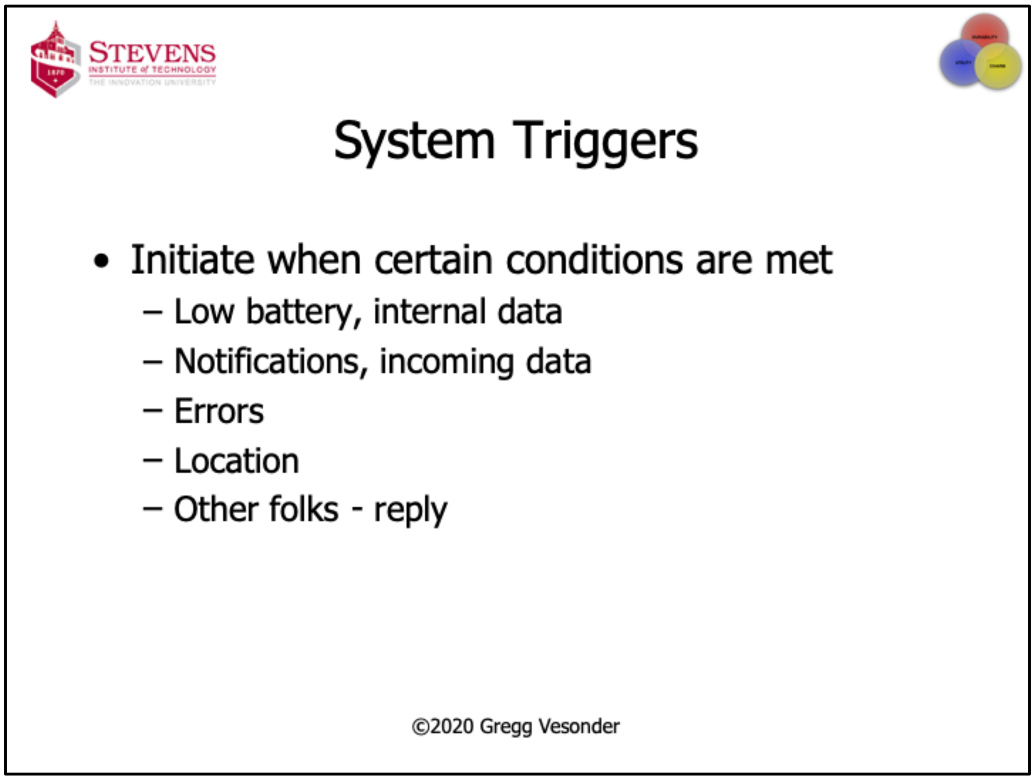
So this is a list of effective triggers – again we are wired to detect motion, make affective use of affordance – the button is screaming to be pushed, knowledge is always good so labeled buttons do help. Salience, a single object on a screen is attention grabbing especially if it stands out form the environment. Invisible triggers are great too – when I say “alexa” to activate my amazon echo.



Invisible triggers

- Should be learnable, always available or available under specific conditions
- Should be guessable or stumbled upon while using other triggers
- For example speech triggers:
  - Always listening – echo, key word trigger + discoverable commands
  - Dialogue
  - Combined with control – Siri
  - For example, gestures, shaking the phone
- Hover, rollover – weather app icon, rolling over gives you temp (bring data forward)

Amazon works hard to educate one about using the echo – each week they discuss functionality in emails and the key word for barge ins alexa is frequently mentioned.

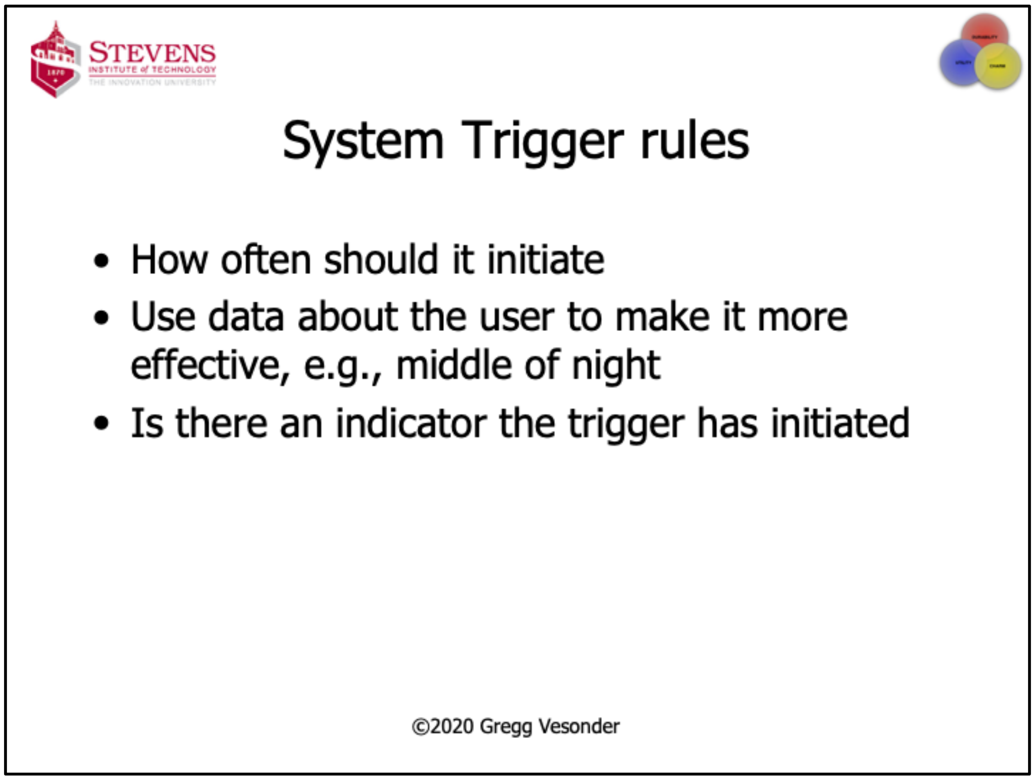


STEVENS INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY

System Triggers

- Initiate when certain conditions are met
  - Low battery, internal data
  - Notifications, incoming data
  - Errors
  - Location
  - Other folks - reply

And sometimes no interaction, adding intelligence to trigger itself is great. Here is a list of such system triggers. Seems that every app uses notification these days. I play the Simpson's game Tapped Out and it is sending me both auditory and printed notifications.



System Trigger rules

- How often should it initiate
- Use data about the user to make it more effective, e.g., middle of night
- Is there an indicator the trigger has initiated

Such automated notifications should not overwhelm the user, rather they should be used judiciously or logged so the user can review them when necessary.