

USER STORY MAPS

The Why's of User Stories

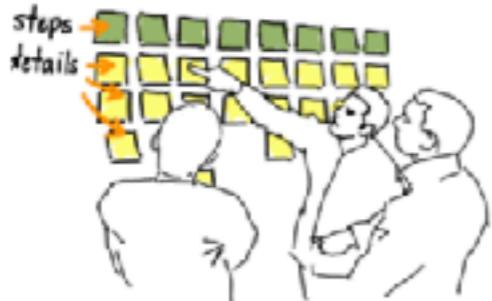
USER STORY MAPS

User Story Mapping - Jeff Patt X

https://jpattonassociates.com/user-story-mapping/

User Story Mapping

Story Mapping is a better way to work with Agile User Stories



User Story Mapping is a dead simple idea. Talk about user's journey through your product building a simple model that tells your user's story as you do. But it turns out this simple idea makes working with user stories in agile development a lot easier.

More importantly it'll keep your users and what they're doing with your product front and center in your products. That's better than getting lost in feature arguments like what normally happens in software development. I'm not making any promises here, but if you use Story Mapping, I believe your products will ultimately be better.

From this page you can get to a number of resources on and off this site that'll help you with story mapping.

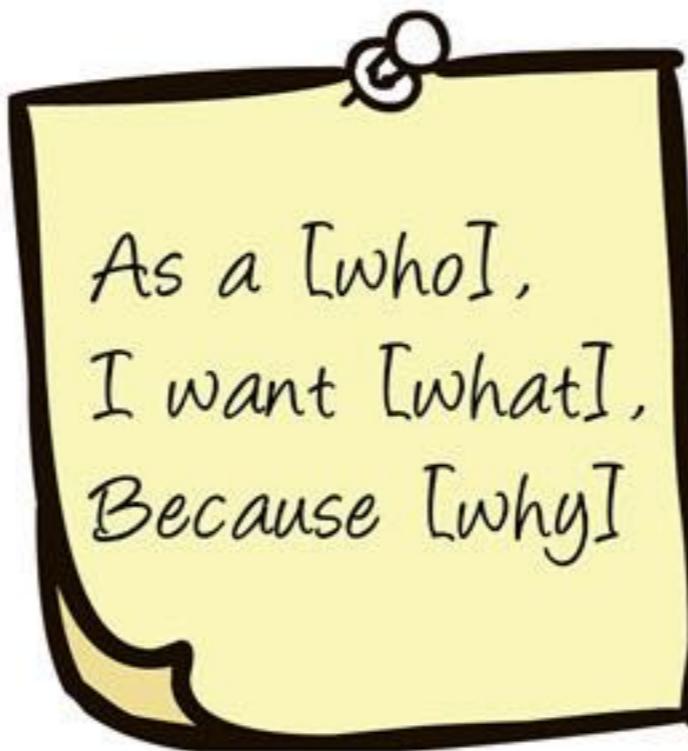
RESOURCE DESCRIPTION



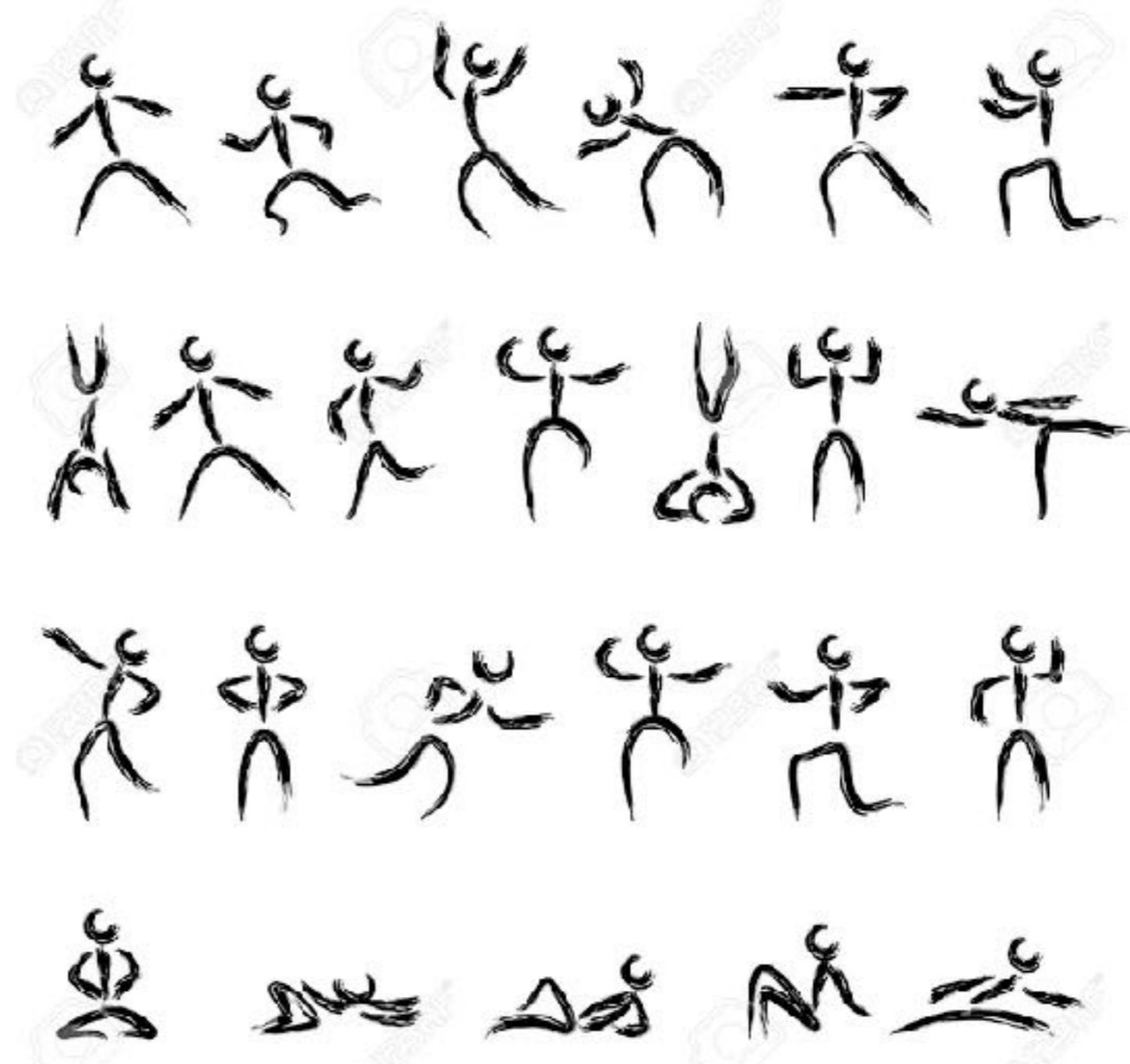
Read This Book: In it, you'll learn about the practice of story mapping and how user stories are really supposed to work in Agile Development. [Read my blog post about the book or just go Buy My Book](#) (Yes, I wrote it).

<https://jpattonassociates.com/wp-content/uploads/2015/01/StoryMapping.png>

USER STORY



USE CASES



FLURPS - REQUIREMENTS

ALL ACRONYMS Any Category

Popular Topics Random Suggest

The screenshot shows the ALLACRONYMS website. At the top, there's a navigation bar with 'ALL ACRONYMS' on the left, a search input field, and category dropdown. Below the navigation is a large blue header with the acronym 'FLURPS' in white. To its right, it says 'means' and lists the six components: Functionality, Localizability, Usability, Reliability, Performance, and Supportability. On the right side of the page, there are two boxes containing Q&A about the acronym. The first box asks 'What does FLURPS stand for?' and the answer is 'FLURPS stands for Functionality, Localizability, Usability, Reliability, Performance, and Supportability'. The second box asks 'What is the abbreviation for Functionality Localizability Usability Reliability Performance Supportability?' and the answer is 'The abbreviation for Functionality Localizability Usability Reliability Performance Supportability is FLURPS'. At the bottom right, it says 'by Oliver Perry | 3 years ago (updated 3 years ago)'

FLURPS

means

Functionality
Localizability
Usability
Reliability
Performance
Supportability

What does FLURPS stand for?

1 Q FLURPS stands for Functionality, Localizability, Usability, Reliability, Performance, and Supportability.

What is the abbreviation for Functionality Localizability Usability Reliability Performance Supportability?

1 Q Functionality Localizability Usability Reliability, Performance Supportability is abbreviated as FLURPS.

by Oliver Perry | 3 years ago (updated 3 years ago)

BUT FIRST! A STORY BEFORE THE STORY!



TropiDoc TropiDoc Resources

WWW.TROPIDOC.ORG WWW.EBS.BU.EDU

Table of Contents

Home Tropi Logos

Command Reference

- All Log Commands
- Basic Input/Output Functions
- Object Functions

Nothing with Date

Process Control

Review Date

Save Data in File Variables

String and Membrane Inputs

Shared Object Inputs

Read, Write, and Print File

Working with Processes

Program Flow

Control the Program Flow

Working with Background Processes

Background Initialization

Debugging

Runtime and Tracing

Print and Assign Tables

Printing and Drawing

ISODAM

Codes

The Single Whitespace

ISODAM

Do-Go Functions

Basic Arithmetic

Accessors

Accessors

BUTIRST (BF)

Reports all but the first element of its input.

Syntax

BF(T1,T2,...,TN)

BF(T1,T2,...,T1)

Description

BUTIRST reports all but the first element of its input. If its input is a list, BUTIRST reports a list containing all but the first element. If its input is a word or number, BUTIRST reports all the characters of the word or number excepting first character. See also **REVERSEOF**, **RELAST**, **FIRST**, and **LAST**.

Example

BUTIRST (MARY HAD A LITTLE LAMB)
Result: ZMM A LITTLE LAMB
BUTIRST (M)
Result: M
BUTIRST (WORLD)
Result: WLD
BUTIRST (1)
Result: 1
BUTIRST (111)
Result: 111



Agenda

But First!

FLURPS

Use Cases

User Story

User Story Maps

Abuse Cases

USER = DEVELOPER

.....

The screenshot shows a web browser window with the title "Programming in BASIC on the TRS-80" and the URL "baugues.com/trs-80". The page content includes a circular profile picture of Greg Baugues, his name, a bio, and links to Email and Twitter. A large image of a TRS-80 terminal screen displaying BASIC code is prominently featured.

Programming in BASIC on the TRS-80

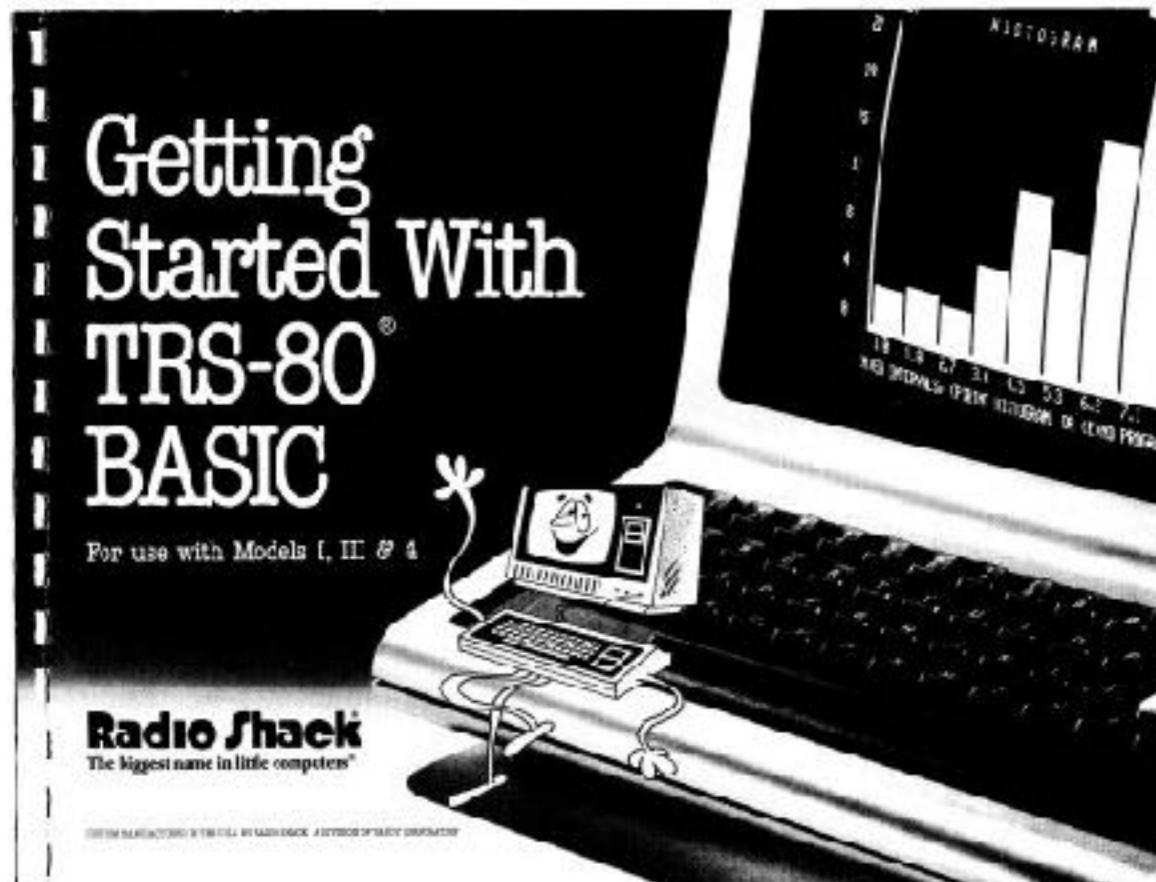
I started programming when I was five years old. If that sounds impressive to you, it's only because you don't realize how easy it was to get started in the '80s.

READY
>10 PRINT "HELLO WORLD"
>20 GOTO

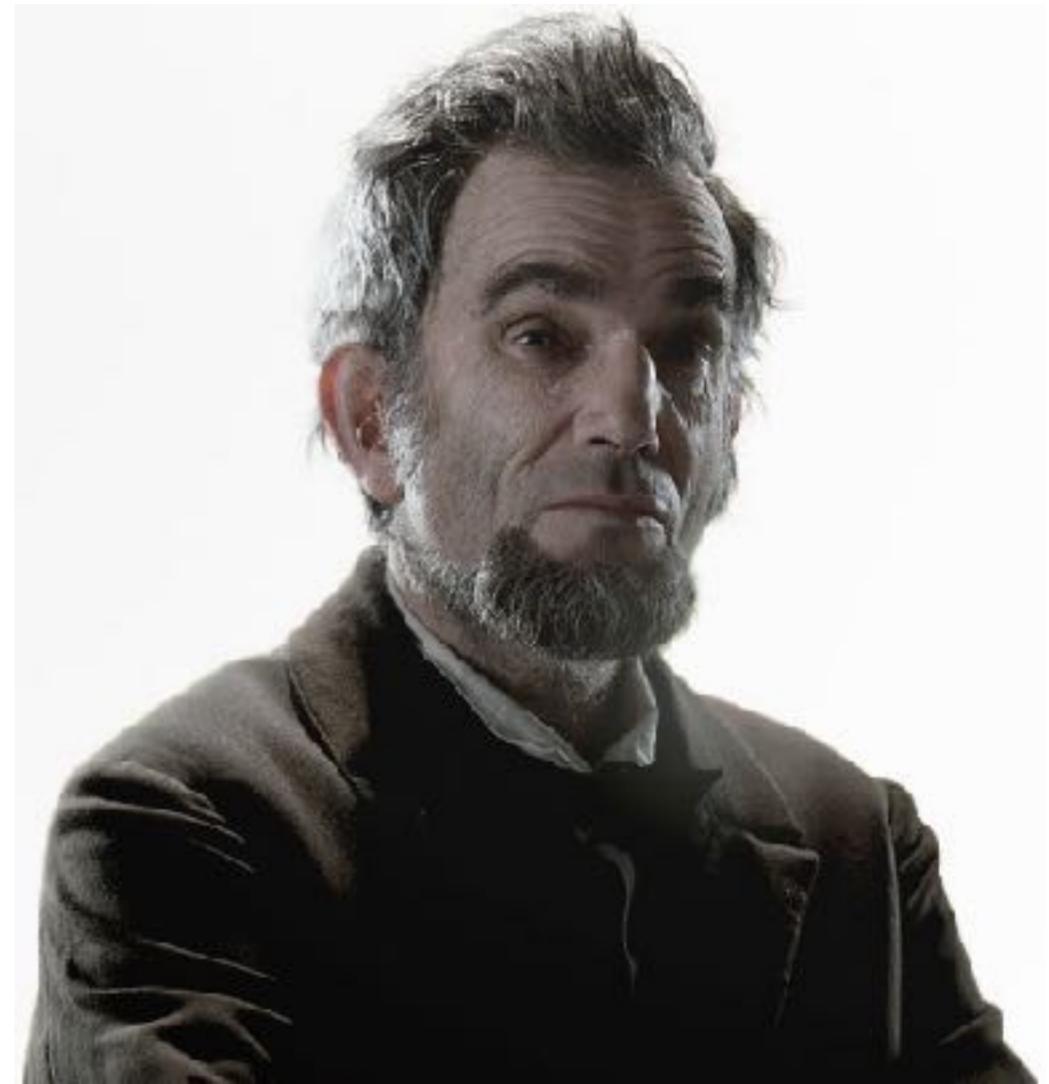
In 1985, my parents bought a Radio Shack TRS-80. I'm not sure why. My dad's a pastor and my mom's a nurse. There was

<http://baugues.com/trs-80>

PONG GAME IN TRS-80 BASIC



WHO IS THIS?



FLURPS - REQUIREMENTS

ALL ACRONYMS Any Category

Popular Topics Random Suggest

The screenshot shows the ALLACRONYMS website. At the top, there's a navigation bar with 'ALL ACRONYMS' (in blue), a search input field, a dropdown for 'Any Category', and a magnifying glass icon. Below the navigation is a large blue header with the acronym 'FLURPS' in white. To the left of 'FLURPS', the word 'means' is followed by a large list of six items: Functionality, Localizability, Usability, Reliability, Performance, and Supportability, all in white text. In the top right corner of the blue header, there's a small box containing the text 'What does FLURPS stand for?'. Below this, there's a post with a blue background and white text. It shows a thumbs-up icon with the number '1', a question mark icon, and the text 'FLURPS stands for Functionality Localizability Usability Reliability Performance Supportability'. Another similar box below it asks 'What is the abbreviation for Functionality Localizability Usability Reliability Performance Supportability?' and provides the same answer. At the bottom right of the page, there's a footer note: 'by Oliver Perry | 3 years ago (updated 3 years ago)'

FLURPS

means

Functionality
Localizability
Usability
Reliability
Performance
Supportability

What does FLURPS stand for?

1 Q FLURPS stands for Functionality Localizability Usability Reliability Performance Supportability

What is the abbreviation for Functionality Localizability Usability Reliability Performance Supportability?

1 Q Functionality Localizability Usability Reliability Performance Supportability is abbreviated as FLURPS

by Oliver Perry | 3 years ago (updated 3 years ago)

FLURPS - REQUIREMENTS + RATIONAL UNIFIED PROCESS

ALL ACRONYMS Any Category

Popular Topics Random Suggest

FLURPS

means

Functionality
Localizability
Usability Reliability
Performance
Supportability

What does FLURPS stand for?

1 Q FLURPS stands for Functionality

Rational Unified Process



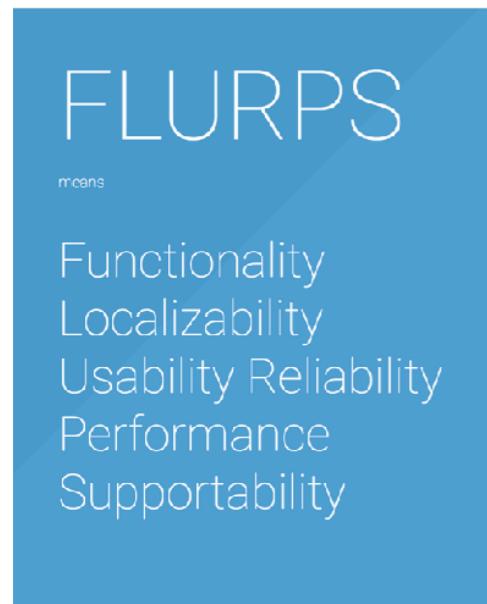
RUP®

Rational Unified Process®

by Oliver Perry | 3 years ago (updated 3 years ago)

FLURPS - REQUIREMENTS & RATIONAL UNIFIED PROCESS

Rational Unified Process



Analysis & Design

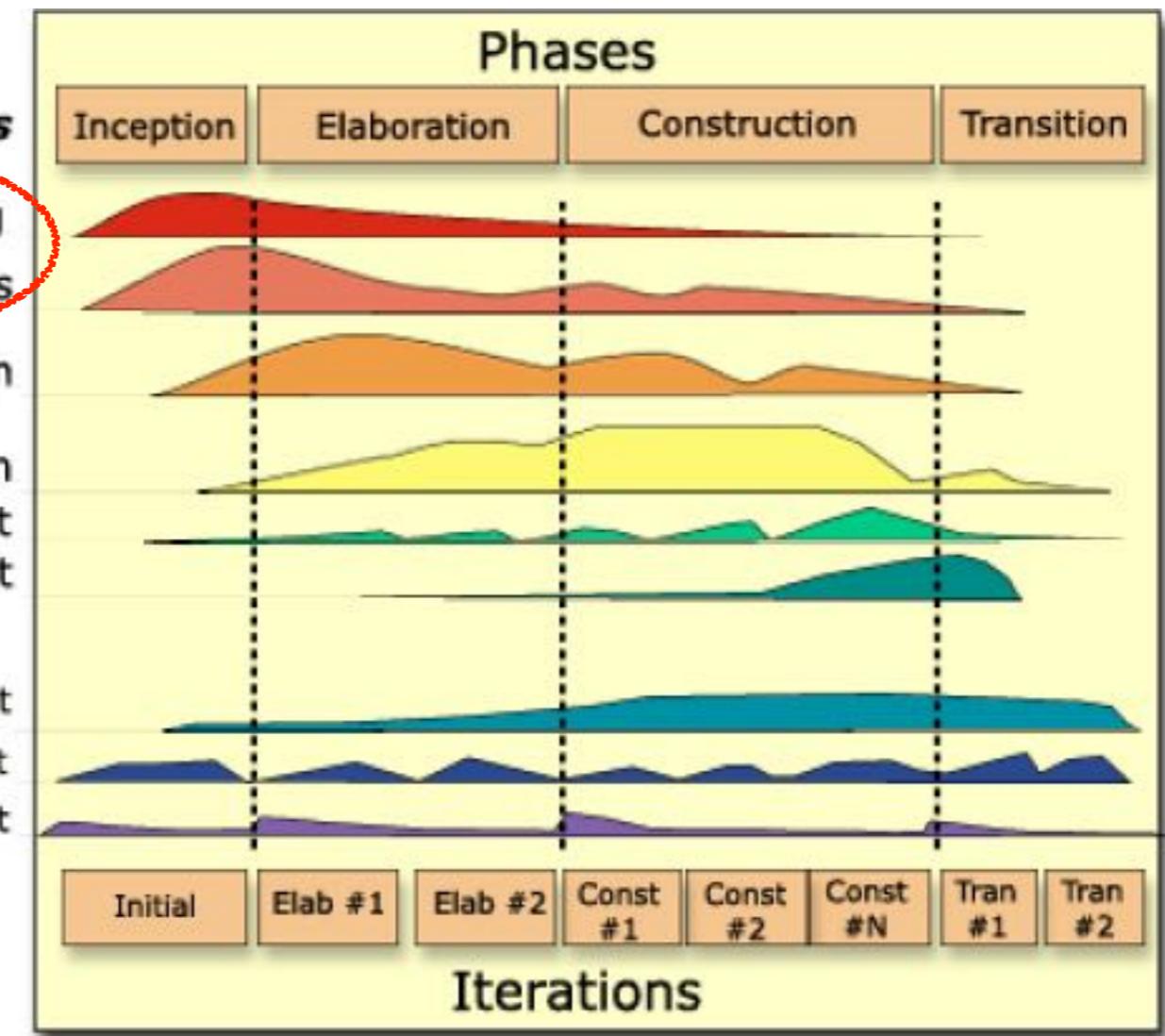
Implementation

Test
Deployment

Configuration
& Change Mgmt

Project Management

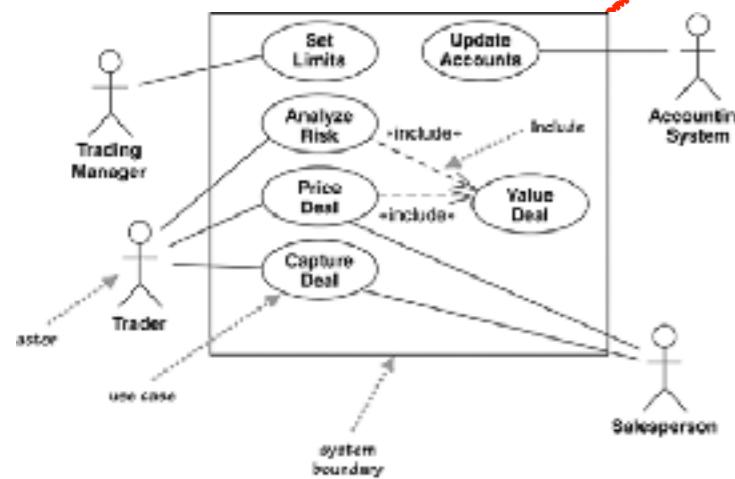
Environment



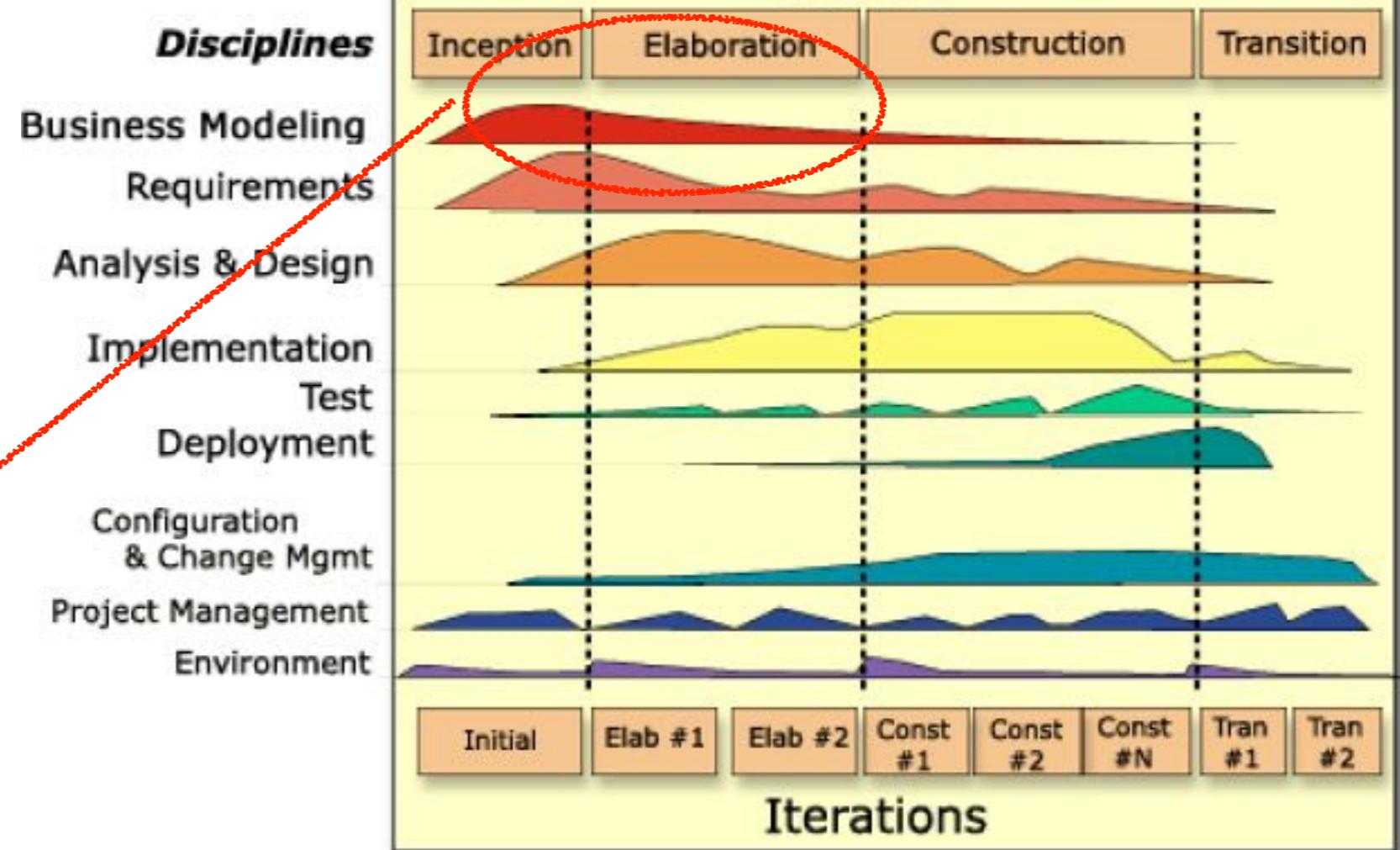
FLURPS - REQUIREMENTS & UML

Use Cases are
Mainly the "F" in
FLURPS (in
practice)

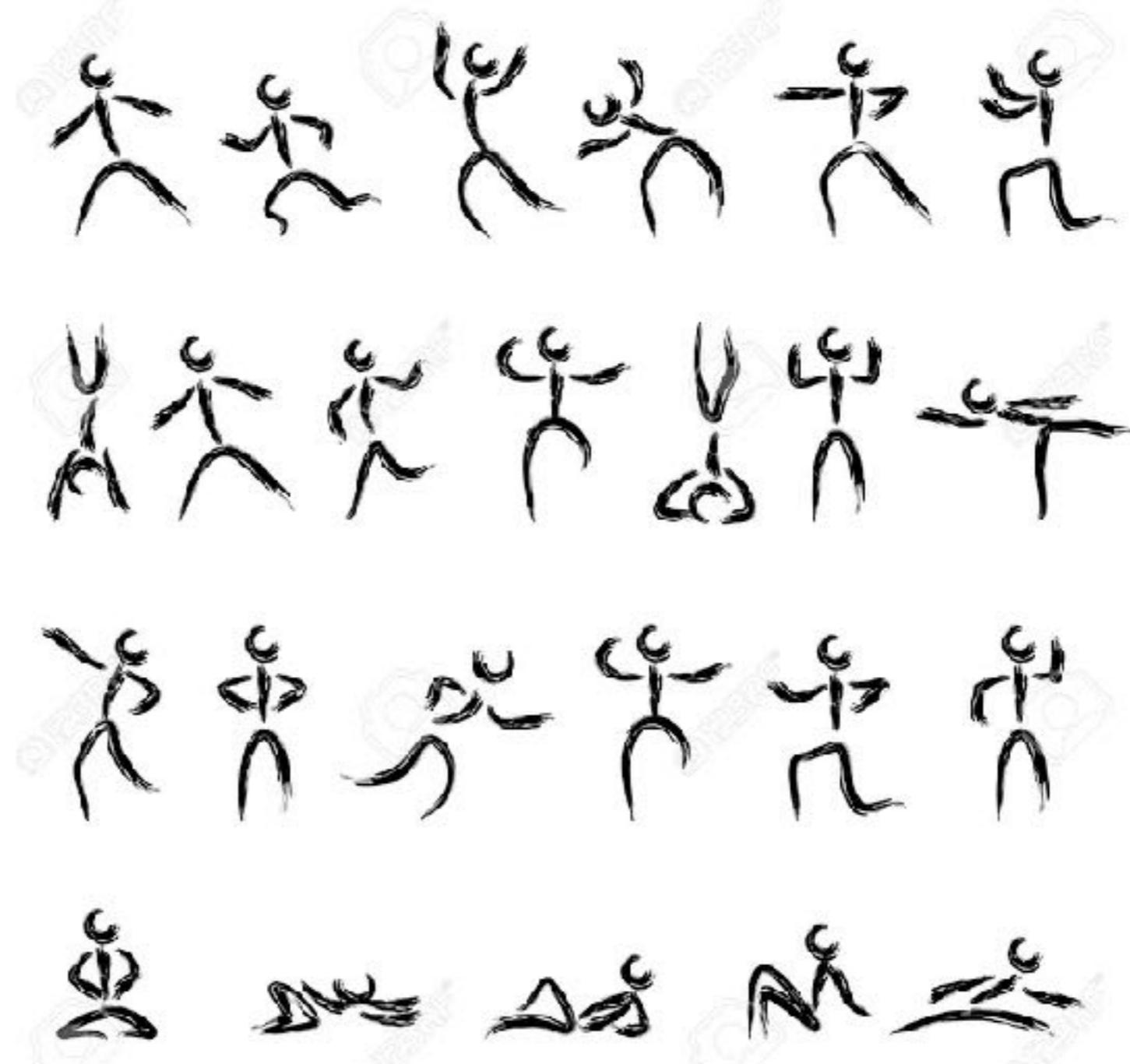
Use Case Diagram



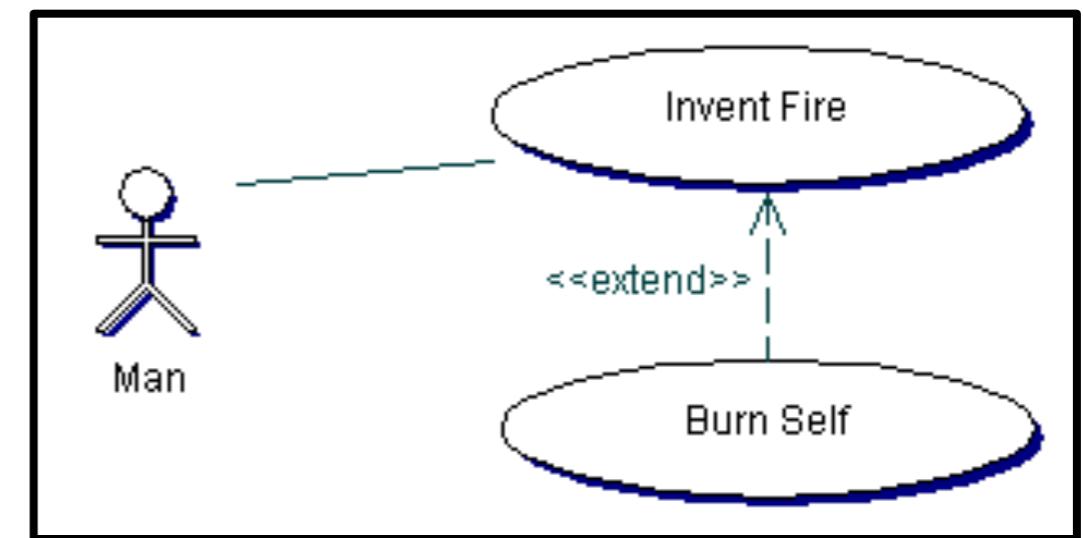
Rational Unified Process



USE CASES



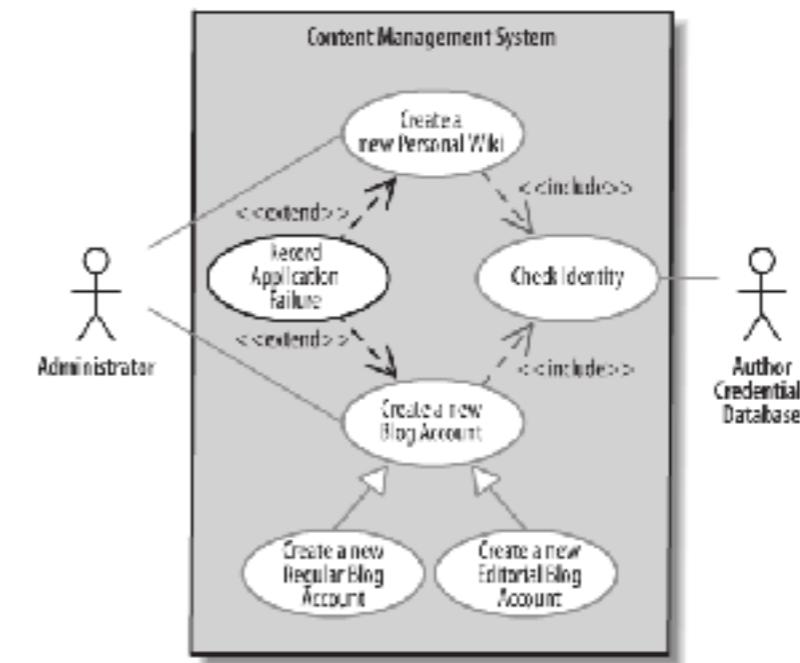
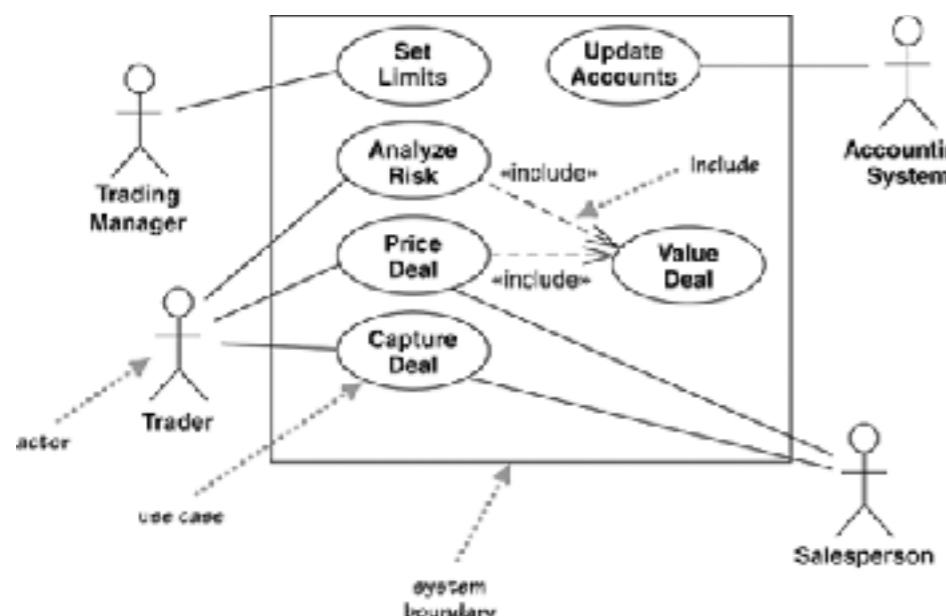
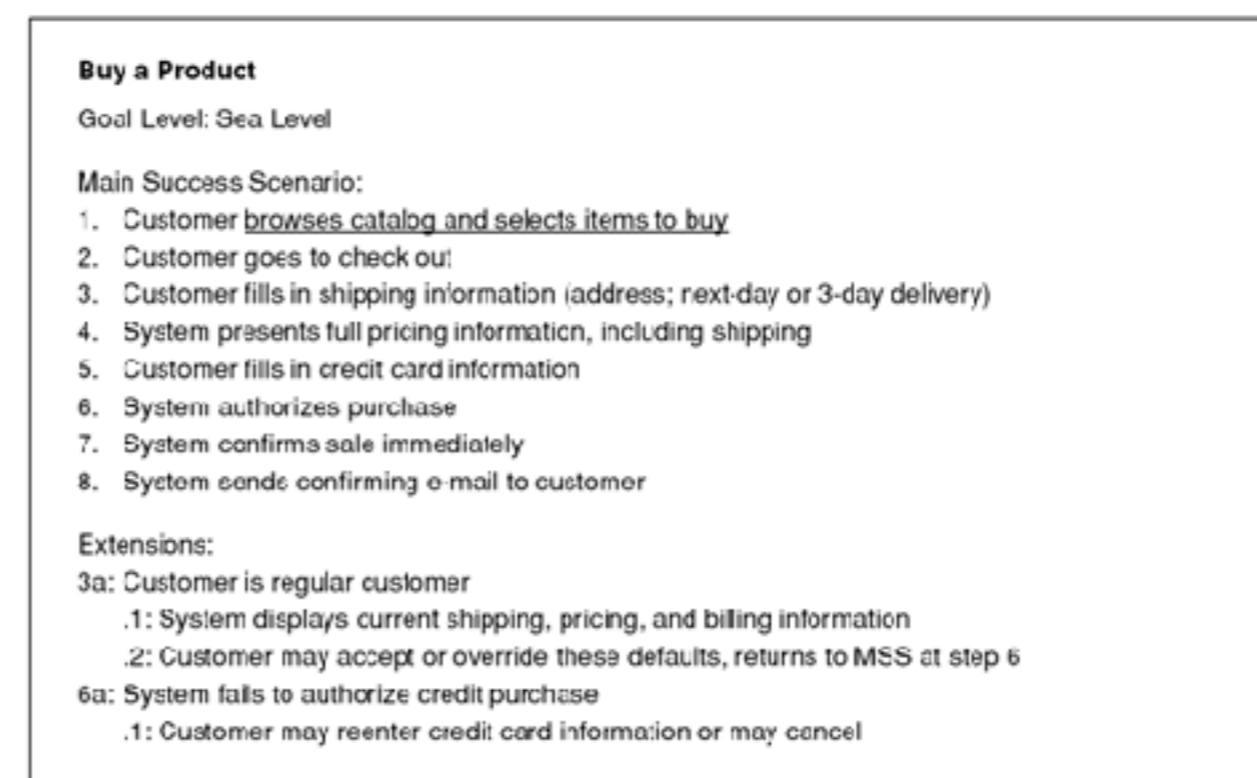
USE CASES – DIAGRAM



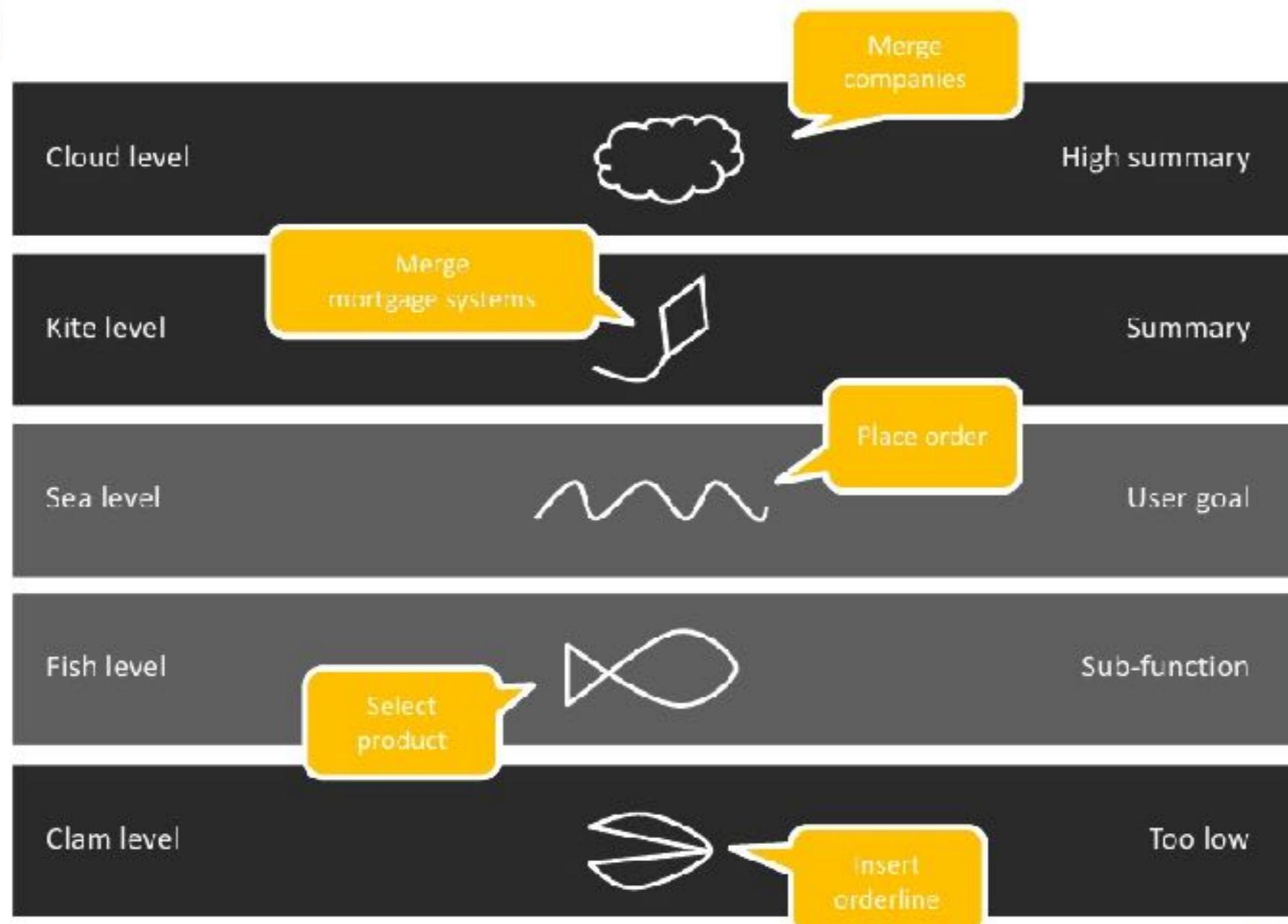
Evolution in Modeling!

USE CASES – NARRATIVE

- Users
 - Actors
 - Users
 - Business
 - System (NFR)
- Goals
 - Something that an *Actor* wants to accomplish in “*a fixed timeframe*”
- Scenarios
 - A Use Case may have multiple scenarios
 - Happy Path
 - Alternate Scenarios
 - Exceptional Scenarios

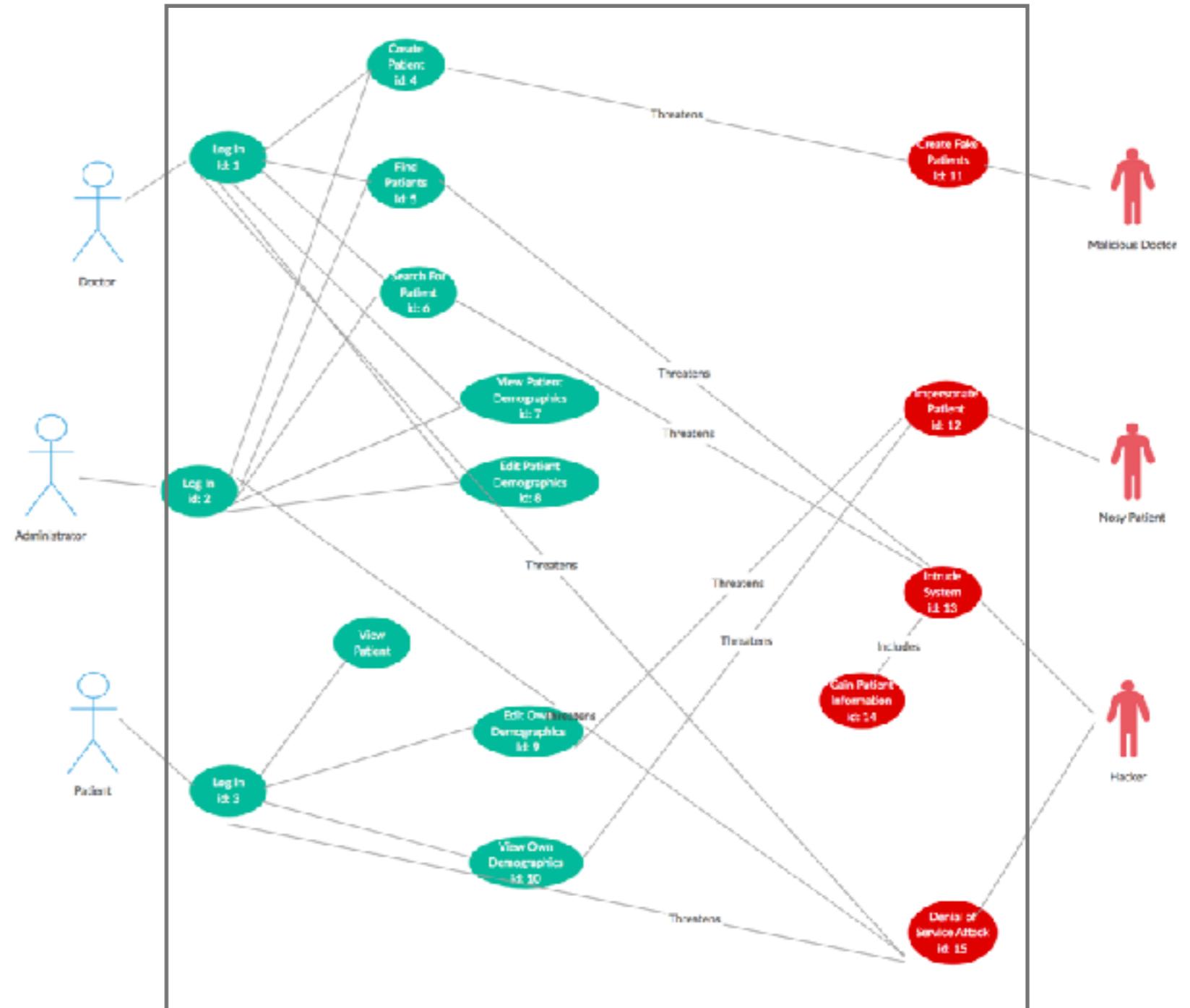


USE CASES - LEVEL OF PERSPECTIVES



ABUSE CASES – BAD ACTORS ARE “USERS” TOO!

- Use Cases vs. Abuse Cases
 - Abuse Case: Actor = Hacker
 - Use Case: Actor = User
- Abuse Case Goals
 - Something the *Hacker* wants to accomplish within a timeframe (or over a period of time)
- Scenarios
 - The “Steps” of the Hack



THE “HACKER’S JOURNEY”

Reconnaissance

Internet Sources

Social engineering

Dumpster diving

Observation

Scanning

Port scanners

Vulnerability scanners

Network mappers

Gaining Access

Abusing a username/password that was found

Exploiting a known vulnerability

Breaking into a weakly secured network

Sending malware to an employee via E-mail or a USB stick on the parking lot

Maintaining Access

Privilege escalation

Installation of a backdoor or remote access trojan

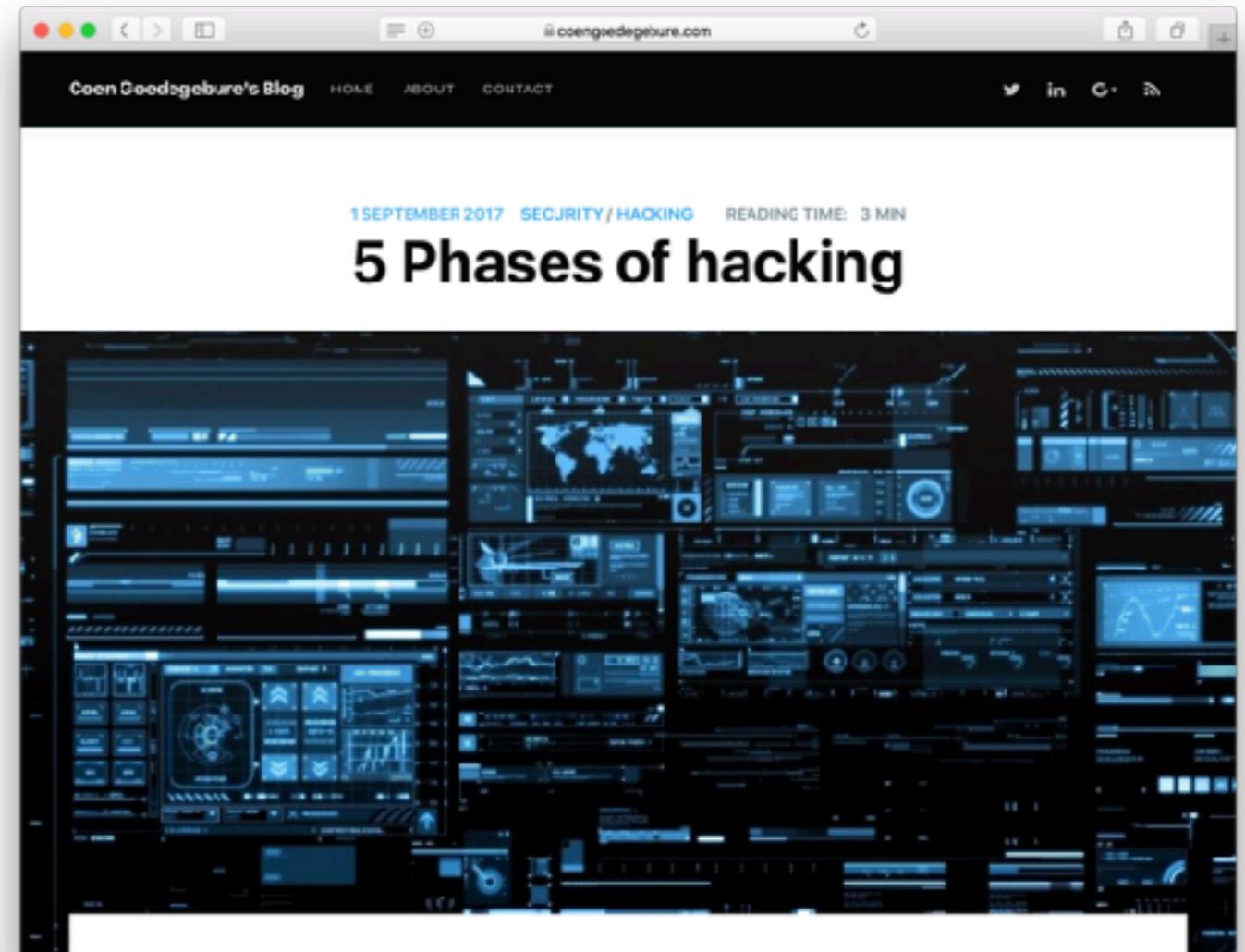
Creating own credentials

Covering Tracks

Remove logging

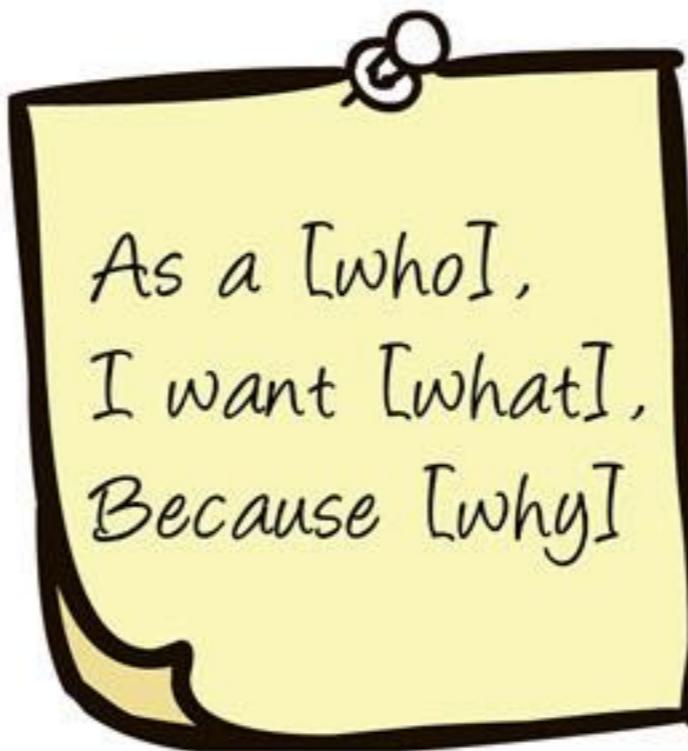
Exfiltration of data via DNS tunneling or steganography

Installation of rootkits



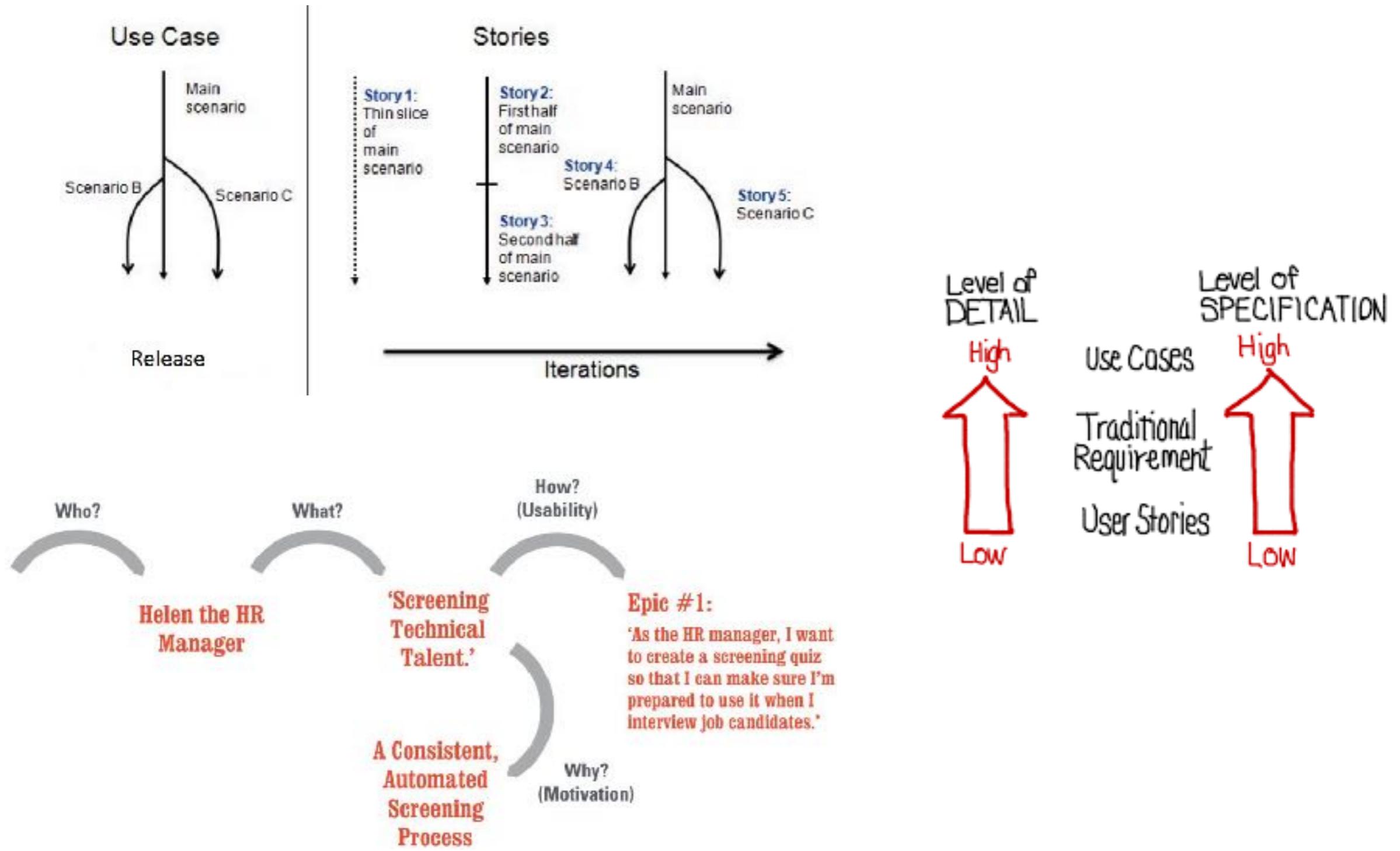
<https://www.coengoedegeure.com/5-phases-of-hacking/>

USER STORY

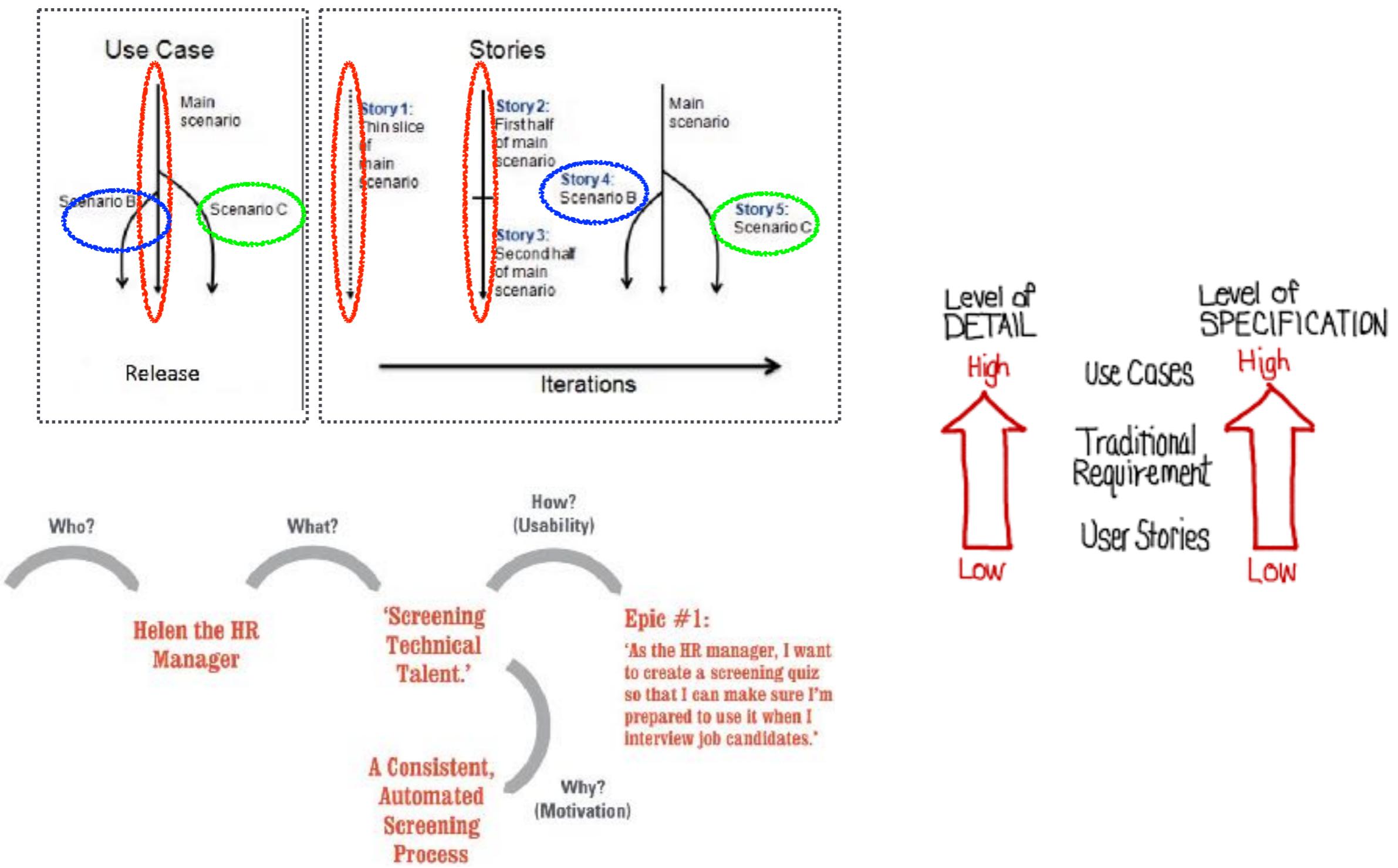


As a [who],
I want [what],
Because [why]

WHAT IS A USER STORY?



WHAT IS A USER STORY?







<http://www.starbucks.com/coffeehouse/mobile-apps>

Mobile Applications

[Tweet](#) [+1](#) [Like](#) [SC](#)



STARBUCKS CARD MOBILE APP
It's the fastest way to pay



[email](#) [share](#) [get code](#) [get transcript](#)



starbucks for Android
Curado for Android is finally here.



starbucks Card Mobile
It's the fastest way to pay.



Mobile Applications
Two iPhone apps from Starbucks

Starbucks Coffee

Create a little extra time in your day,

with the Starbucks card mobile app.

It's the fastest way to pay for your drink.

View and reload your Starbucks card balance,

and even check your My Starbucks Rewards Stars.

So now you'll have more time to relax.

Get to work,
walk the dog,

or just enjoy your coffee even more.

Just scan
and go.

Starbucks card mobile app,
it's the fastest way to pay.

Starbucks Coffee
Get it for iPhone or Blackberry.

User Story

As a customer with the Starbucks card mobile app,
I can view, reload, check my rewards and pay on the go,
so that I can have a little extra time in my day to relax,
get to work faster and walk my dog.

THERE THREE “C’S” OF USER STORIES

Stories are for telling

In the late 1990's Kent Beck had a simple idea to solve one of the biggest challenges in software development: communicating the details of what to build. By simply getting together and "telling our stories" we could build shared understanding in the minds of everyone involved.

In the conversation we'd focus not only on what to build, but who would use the software and why. Our goal is to identify the most valuable thing we could most economically build.

Stories get their name from how they're used and not how they're written



Kent Beck, author of Extreme Programming Explained

What I was thinking of was the way users sometimes tell stories about the cool new things the software they use does.

"I type in the zip code and it automatically fills in the city and state without me having to touch a button."

I think that was the example that triggered the idea. If you can tell stories about what the software does and generate energy and interest and a vision in your listener's mind, then why not tell stories before the software does it?



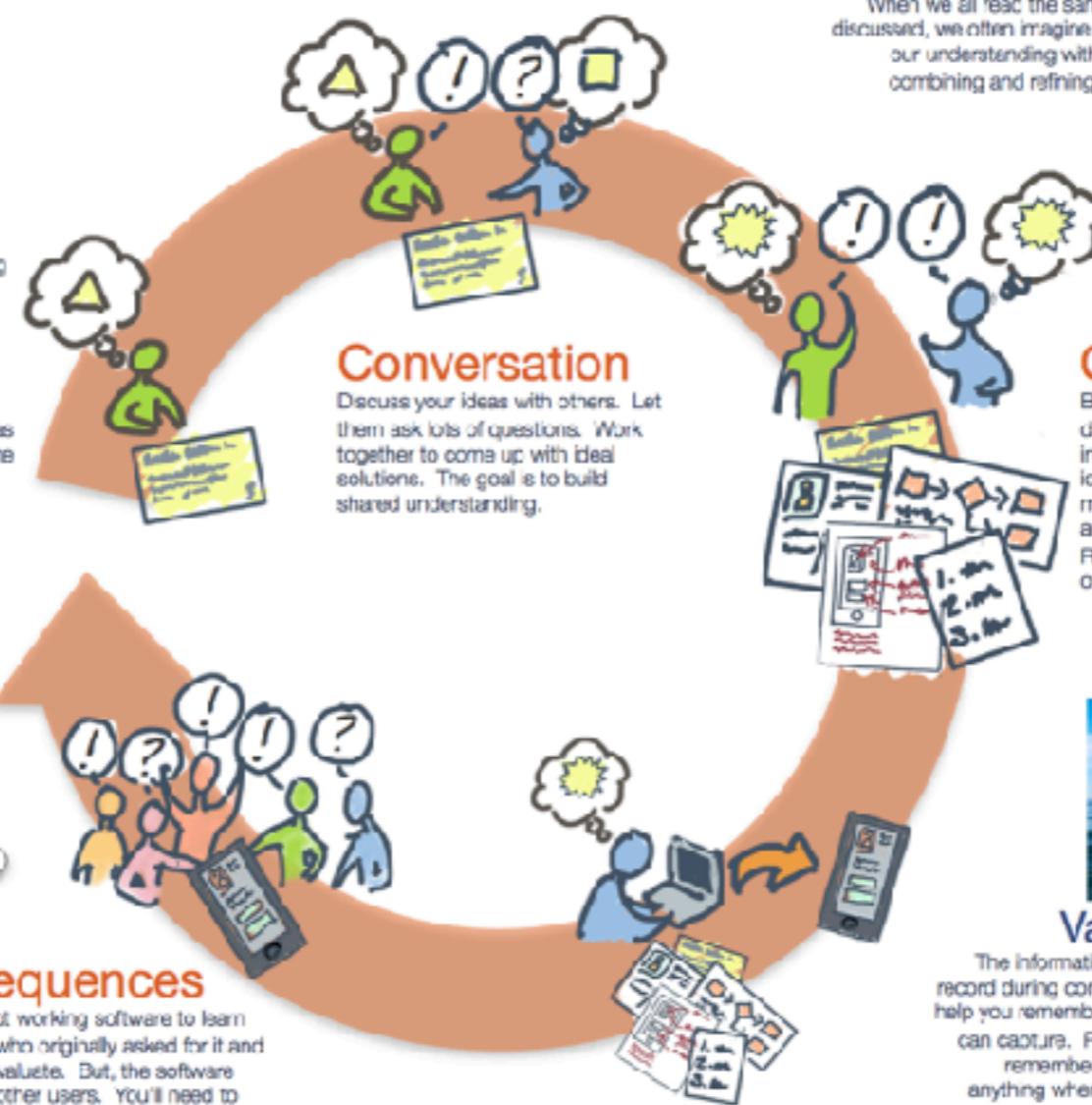
For every story in your backlog put in three cards. The first is what you want, the second one is there to remind you to fix the first one. The third is to remind you to fix it again. You've got to iterate or you're not doing it right.



Shared Understanding

When we all read the same document or hear the same discussion, we often imagine different things. It's describing our understanding with words and pictures, and then combining and refining our ideas that leads to shared understanding.

Shared documents are not shared understanding



Vacation Photos

The information, drawings, and models you record during conversations are mementos that help you remember many more details than you can capture. People that weren't there won't remember – just like they wouldn't recall anything when seeing your vacation photos.

Construction

Developers, testers, and others equipped with information from conversations and the shared understanding that comes with it build and test the software.

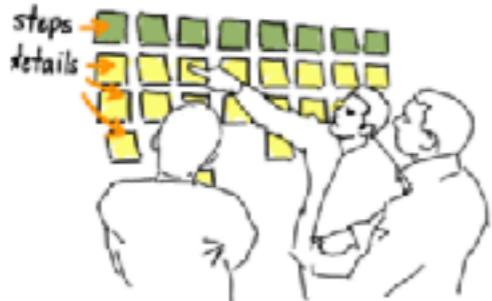
USER STORY MAPS

User Story Mapping - Jeff Patt X

https://jpattonassociates.com/user-story-mapping/

User Story Mapping

Story Mapping is a better way to work with Agile User Stories



User Story Mapping is a dead simple idea. Talk about user's journey through your product building a simple model that tells your user's story as you do. But it turns out this simple idea makes working with user stories in agile development a lot easier.

More importantly it'll keep your users and what they're doing with your product front and center in your products. That's better than getting lost in feature arguments like what normally happens in software development. I'm not making any promises here, but if you use Story Mapping, I believe your products will ultimately be better.

From this page you can get to a number of resources on and off this site that'll help you with story mapping.

RESOURCE DESCRIPTION

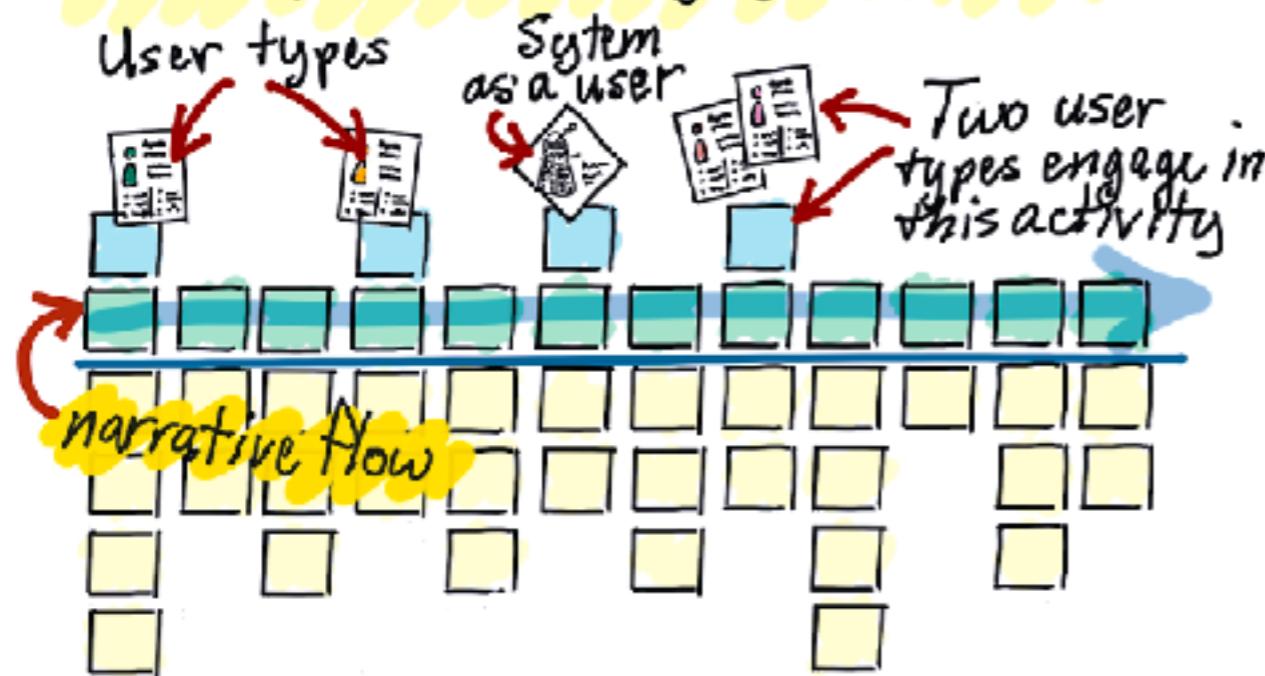


Read This Book: In it, you'll learn about the practice of story mapping and how user stories are really supposed to work in Agile Development. [Read my blog post about the book or just go Buy My Book](#) (Yes, I wrote it).

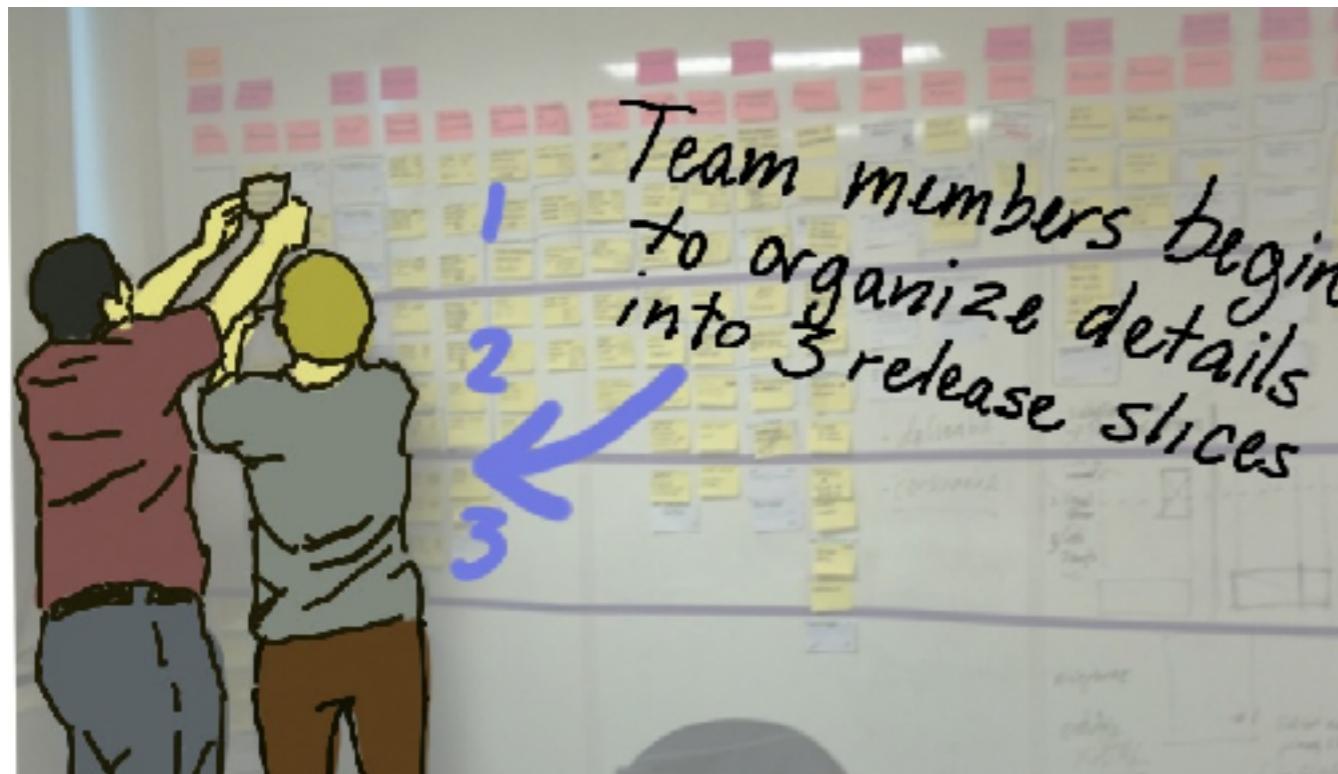
<https://jpattonassociates.com/wp-content/uploads/2015/01/StoryMapping.png>

USER STORY MAPS - THE USER'S JOURNEY

This map crosses many types of users



Adding risk stories to make risk visible



STORY MAP CONCEPTS

Users

A map tells a story about a type of person doing something to reach a goal. Make sure to include them in your map along with a little information about them.

Try using lightweight persona sketches to describe your users.

Backbone

Activities and tasks at a higher goal level give the story map its structure. The backbone is arranged in a narrative flow. Smaller sub-tasks, details and variations hang down to form the ribs connected to the backbone.



Release Slice

Use a tape line to identify slices of tasks that users might use your software for to reach their goals. The smallest number of tasks that allow your specific target users to reach their goal compose a viable product release.

Use release slices to identify small experiments, minimal viable product releases, or a "walking skeleton" version of your product.

Identify the target outcomes of your slice in a sticky note or card to the left of the slice.

User Tasks

User's tasks are short verb phrases that are the basic building block of a map. If I ask you what you did earlier today when using email, you'll likely respond with tasks like:

- Read an email message
- Respond to a message
- Mark a message as spam

Goal-Level

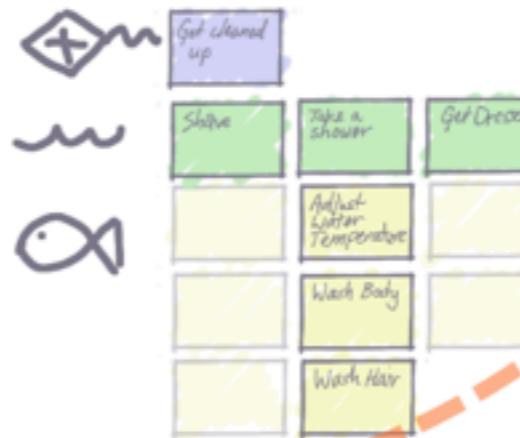
The actions that users take in order to reach their larger goals have a goal level themselves that's tied to user behavior.

Summary: lots of tasks done in support of a bigger goal.

Functional: I'd expect to complete this task before taking a break.

Sub-Functional: smaller things done in support of a bigger tasks.

As you read across tasks in the backbone, check to make sure that tasks are of a similar goal level.



Activities

Activities organize tasks done by similar people at similar times to reach a goal. For your email software activities might include

- Going through my inbox
- Configuring my email client
- Organizing messages into folders

Narrative Flow

The left to right axis in a story map is organized in the order you'd tell the story about your user to someone else.

Of course any specific user might choose to do different things in a different order. Use conversation to explain the details and variations.

If you're looking for the precision of a workflow model, flow chart, or UML model, then a story map isn't your best choice.

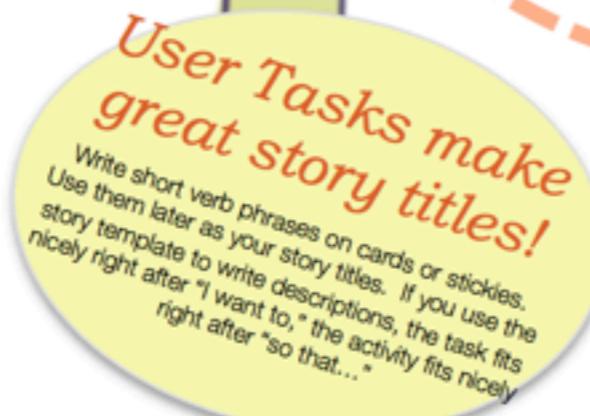
A story map will take lots of conversation to use effectively. But then that's the purpose of stories.

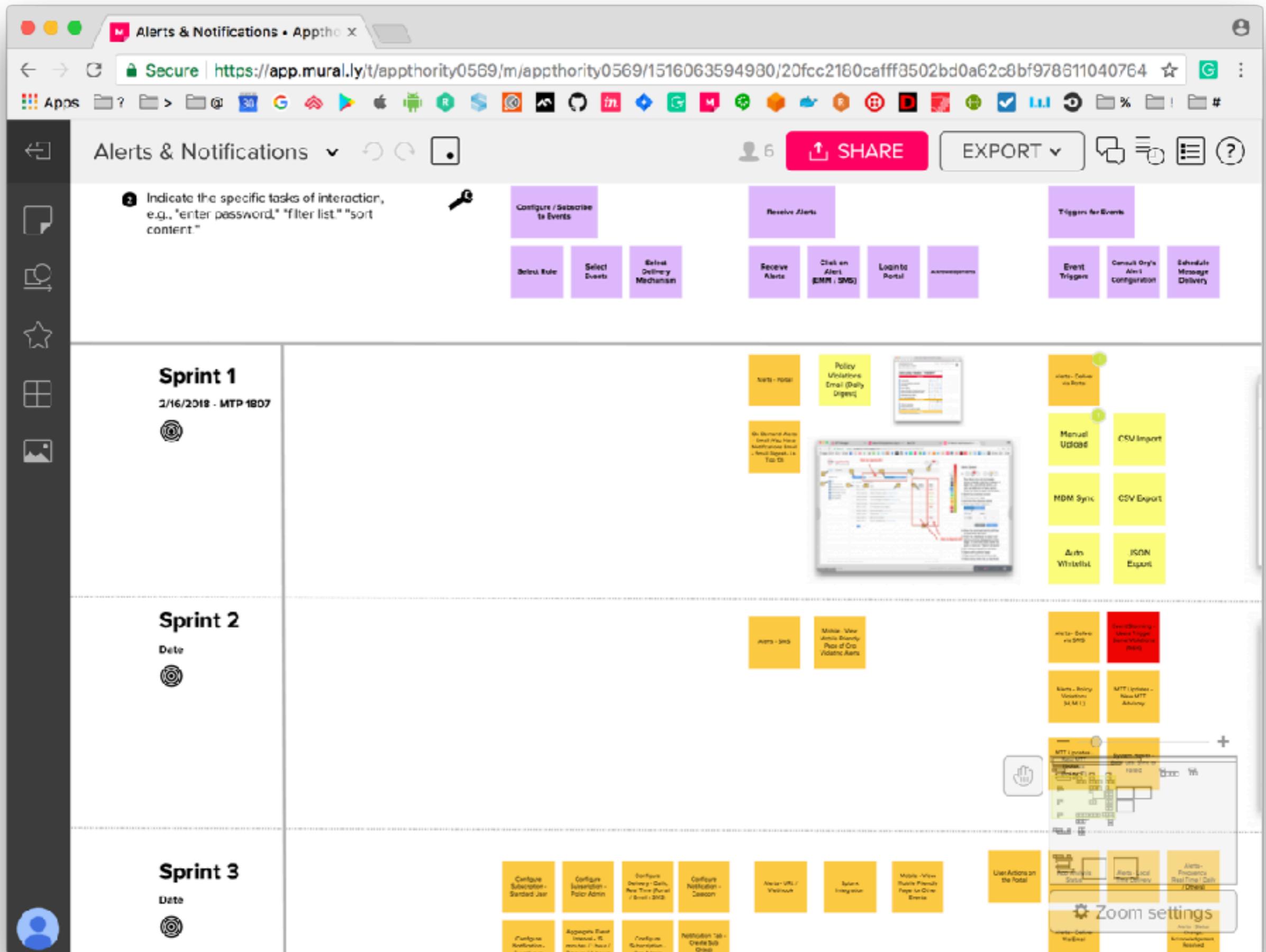
Details, Details...

Break down high goal level tasks into:

- Sub-tasks
- Alternative tasks
- Exceptions
- Details

Down in the details of the map, it's OK to include details about what UI might look like or what the system might do in the background.





THE HACKER'S JOURNEY

Hacking Phases: Reconnaissance, Scanning, Gaining Access, Maintaining Access, Covering Tracks

1 User Journey

- Add the broader phases of interaction the user has with the product or service, e.g. "log in," "search," "view content."
- Indicate the specific tasks of interaction, e.g., "enter password," "filter list," "sort content."



Release 1

Date
●



Release 2

Date
●



Threat Indicators that addresses the Hacks.

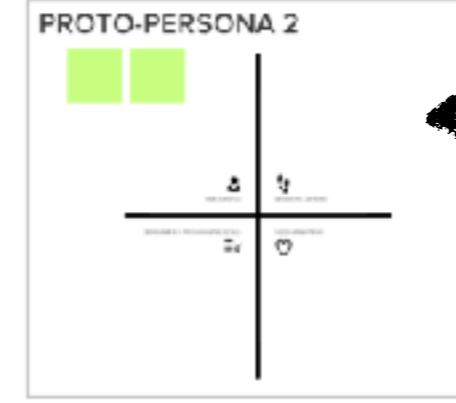
2 Personas



Create descriptions of your users.

- As a team, determine the key users who will interact with the product. Then create the proto-personas by using the personas of the framework.
- Discuss how the system will address their needs and pain points.

IMPORTANT: If you already have personas, combine the frameworks to the right and add them there.



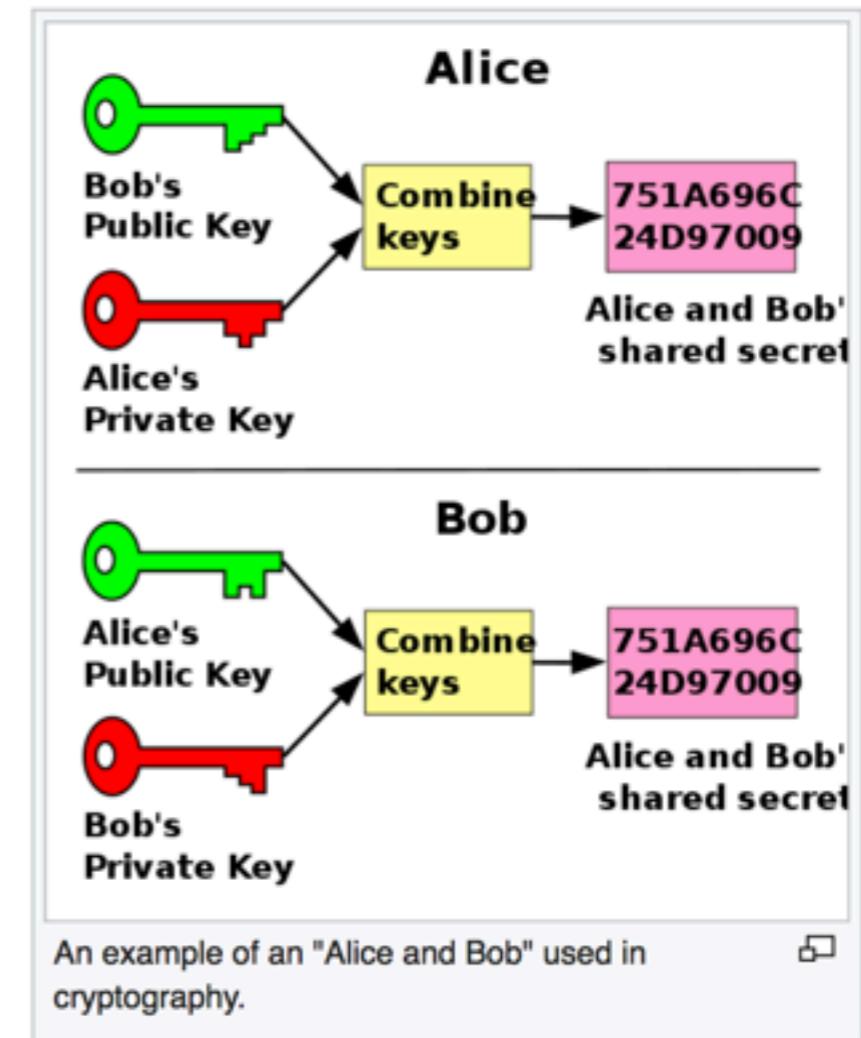
Security Players "Personas"

USER PERSONAS

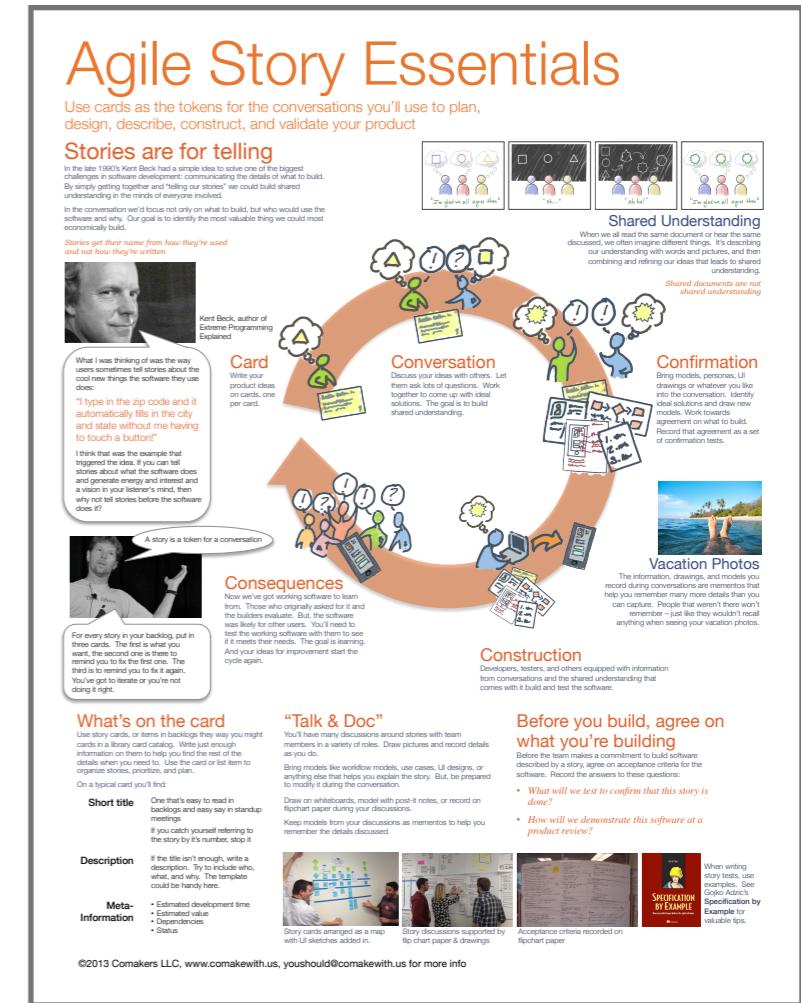
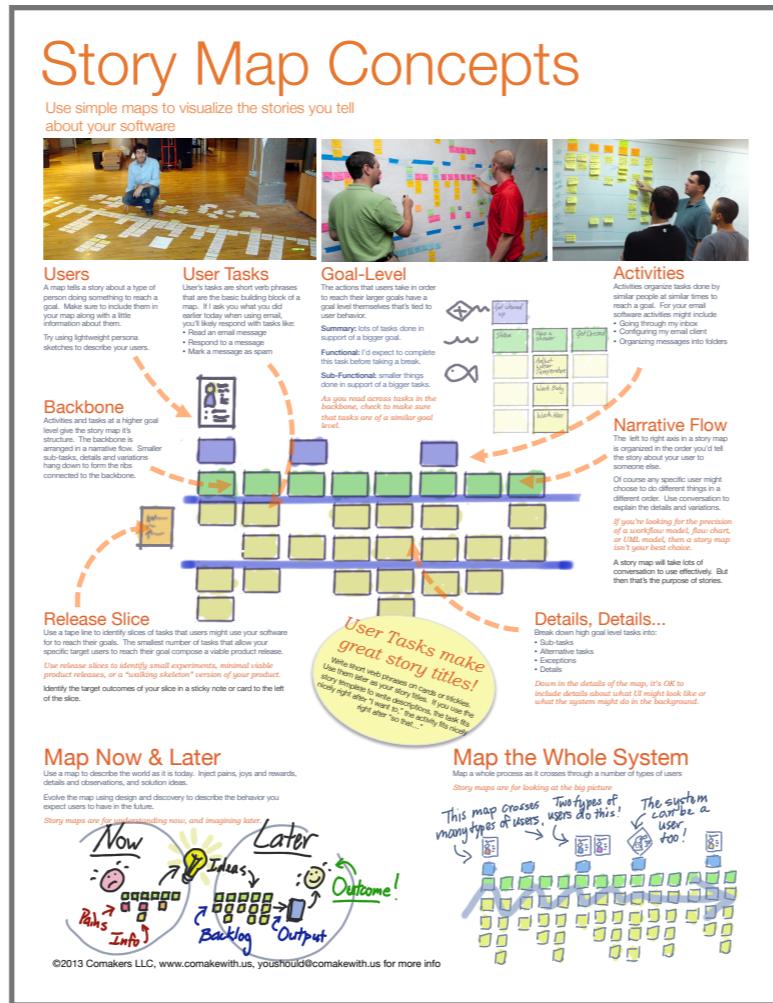
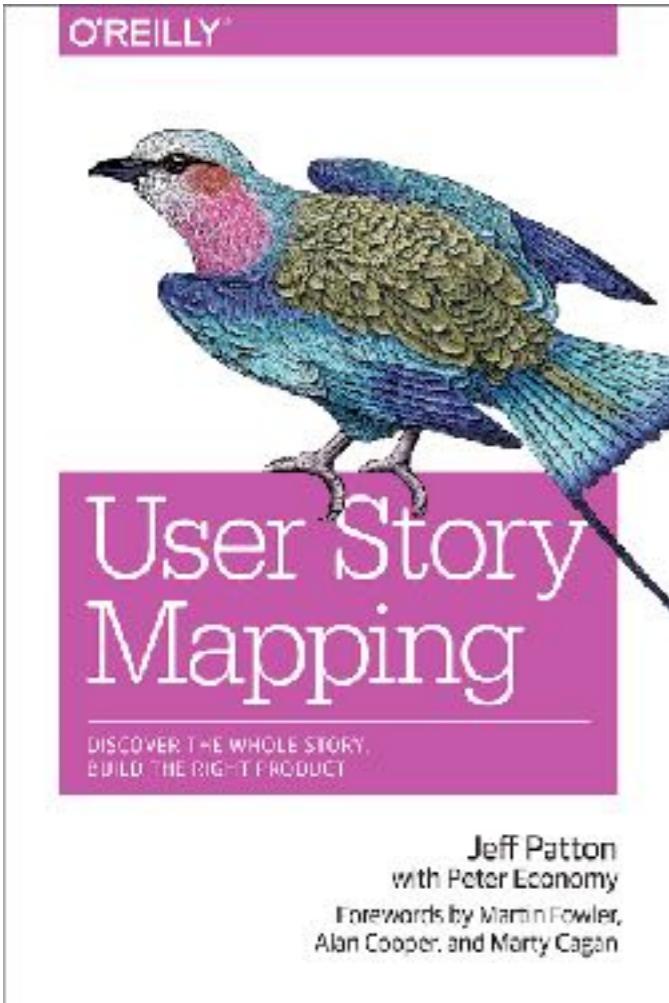
Cast of characters [edit]

The most common characters are Alice and Bob. Eve, Mallory, and Trent are also common names, and have fairly well-established "personalities" (or functions). The names often use rhyming mnemonics (for example, Eve, "eavesdropper;" Mallory, "malicious"). Other names are much less common, and flexible in use.

- *Alice* and *Bob*. The original, generic characters. Generally, Alice and Bob want to exchange a message or cryptographic key.
- *Carol, Carlos or Charlie*. A generic third participant.
- *Chuck*. A third participant, usually of malicious intent.^[11]
- *Craig*. A *password cracker*, often encountered in situations with stored passwords.
- *Dan, Dave or David*. A generic fourth participant.
- *Erin*. A generic fifth participant, but rarely used, as "E" is usually reserved for Eve.
- *Eve*. An *eavesdropper*, who is usually a passive attacker. While she can listen in on messages between Alice and Bob, she cannot modify them. In *quantum cryptography*, Eve may also represent the *environment*.
- *Faythe*. A *trusted advisor*, courier or intermediary. Faythe is used infrequently, and is associated with Faith and Faithfulness. Faythe may be a repository of key service or courier of shared secrets.^[citation needed]
- *Frank*. A generic sixth participant.



REFERENCE



<https://jpattonassociates.com/user-story-mapping/>

“AS-IS” VS. “TO-BE” — CHANGE THE WORLD!

