Crafting Software Requirements & Picturing Architecture & Design

ENSE 281 with Tim Maciag

University of Regina

FACULTY OF ENGINEERING & APPLIED SCIENCE
Software Systems Engineering

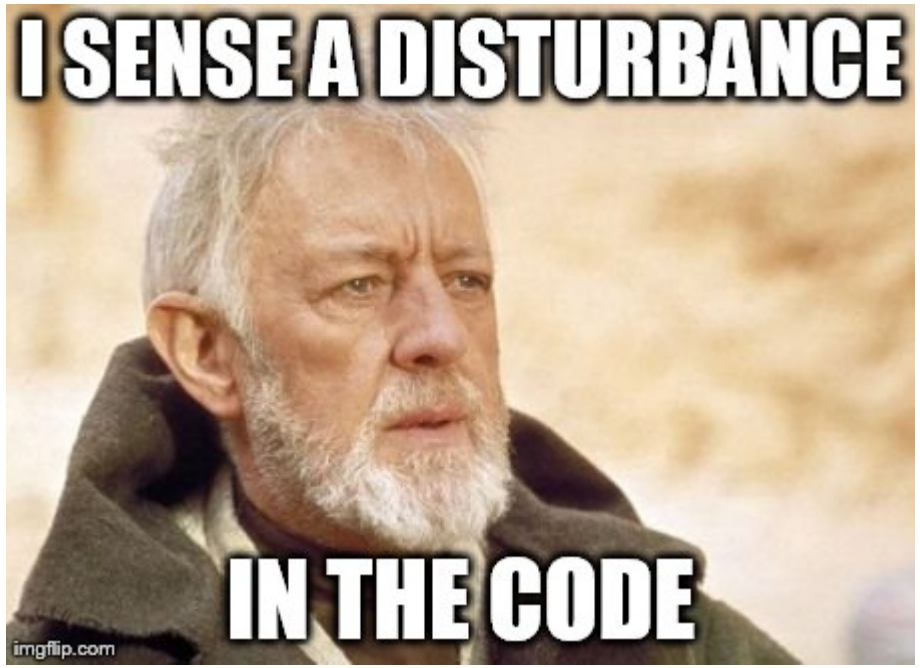
CC BY SA

Outline

- Requirements and problem prerequisites
- Picturing architecture
- Unified Modelling Language (UML)
- Use case diagramming
- Activity diagramming
- Interaction diagramming
- Data diagramming
- Class/CRC diagramming
- Picturing design (we already know how to do this one!)

Requirements digging & prerequisites

- Recall: Software construction (programming, debugging) is really the only given in any software project

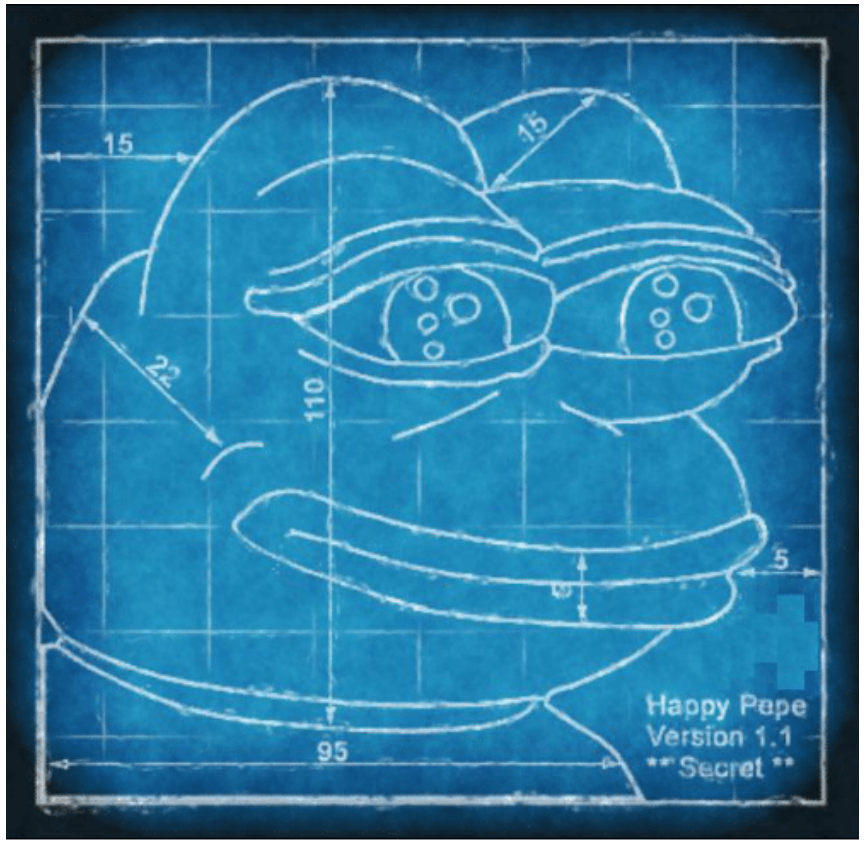


Requirements digging & prerequisites

- Recall: Software construction (programming, debugging) is really the only given in any software project
- The old adage: “Measure twice, cut once”

Requirements digging & prerequisites

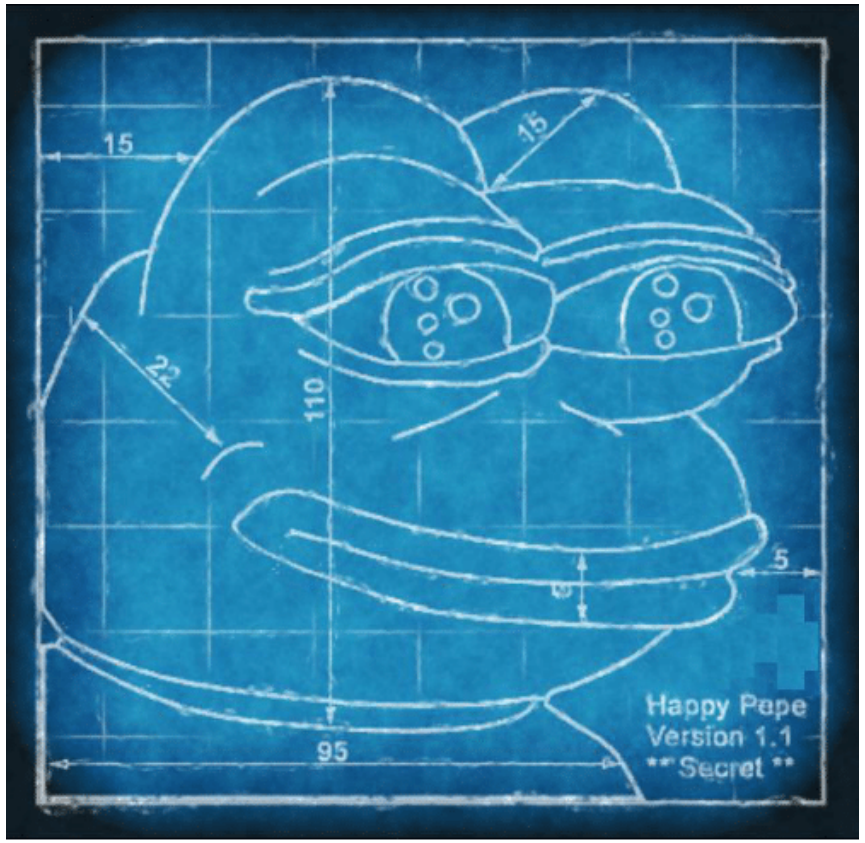
- Recall: Software construction (programming, debugging) is really the only given in any software project
- The old adage: “Measure twice, cut once”
- Prep work for software construction



Rare Blueprint Pepe

Requirements digging & prerequisites

- Recall: Software construction (programming, debugging) is really the only given in any software project
- The old adage: “Measure twice, cut once”
- Prep work for software construction
- Benefits



Rare Blueprint Pepe

Cost of defects

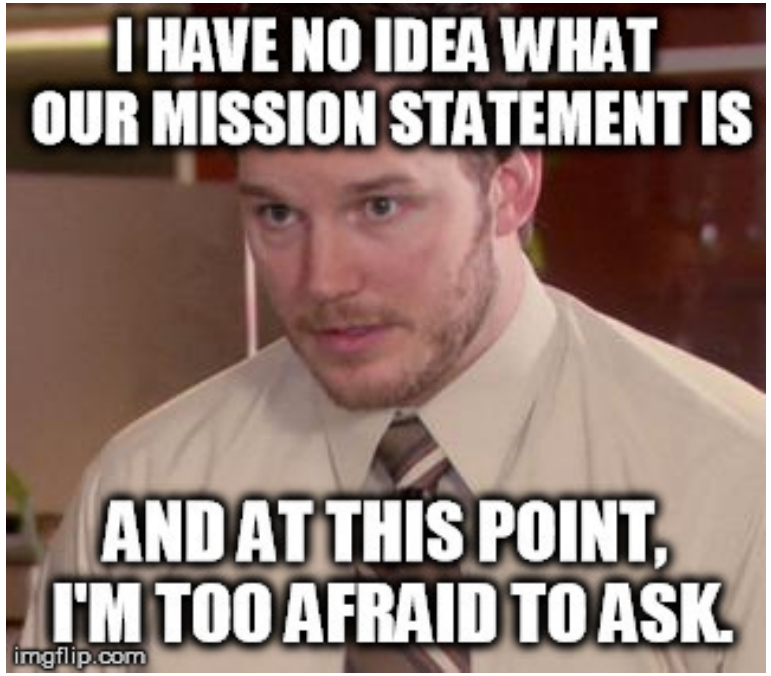
	Time Detected				
Time Introduced	Requirements	Architecture	Construction	System Test	Post-Release
Requirements	1	3	5–10	10	10–100
Architecture	—	1	10	15	25–100
Construction	—	—	1	10	10–25

Source: Adapted from "Design and Code Inspections to Reduce Errors in Program Development" (Fagan 1976), *Software Defect Removal* (Dunn 1984), "Software Process Improvement at Hughes Aircraft" (Humphrey, Snyder, and Willis 1991), "Calculating the Return on Investment from More Effective Requirements Management" (Leffingwell 1997), "Hughes Aircraft's Widespread Deployment of a Continuously Improving Software Process" (Willis et al. 1998), "An Economic Release Decision Model: Insights into Software Project Management" (Grady 1999), "What We Have Learned About Fighting Defects" (Shull et al. 2002), and *Balancing Agility and Discipline: A Guide for the Perplexed* (Boehm and Turner 2004).

- S. McConnell's. Code Complete, Second Edition. Microsoft Press. 2004

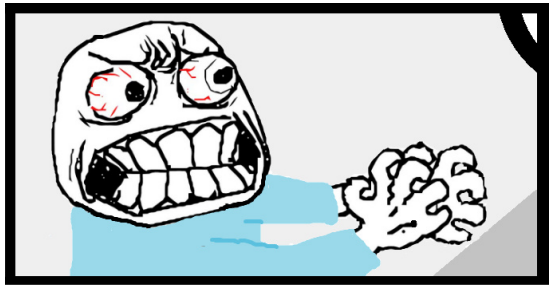
Problem-definition prerequisite

- Also called product vision, vision statement, mission statement, and product definition
  - As in the Business Case! – Hey, you know this!



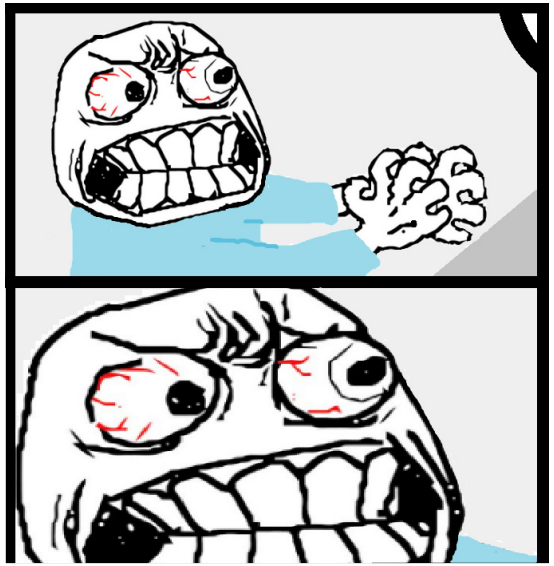
Problem-definition prerequisite

- Also called product vision, vision statement, mission statement, and product definition
  - As in the Business Case! – Hey, you know this!
- Defines the problem, not the solution



Problem-definition prerequisite

- Also called product vision, vision statement, mission statement, and product definition
  - As in the Business Case! – Hey, you know this!
- Defines the problem, not the solution
- Written from the user's point of view, in the user's language

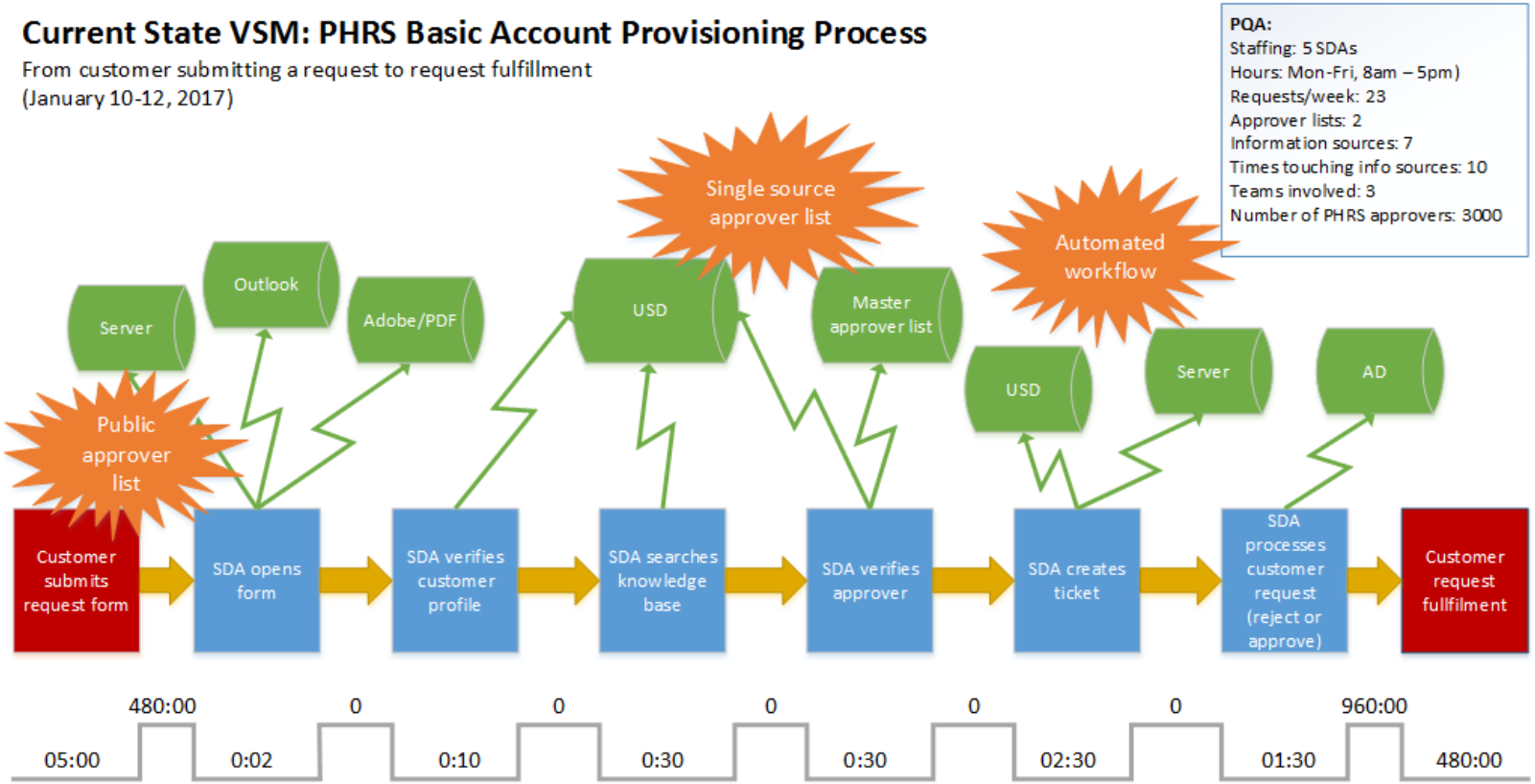


Problem-definition prerequisite

- Also called product vision, vision statement, mission statement, and product definition
  - As in the Business Case! – Hey, you know this!
- Defines the problem, not the solution
- Written from the user's point of view, in the user's language
- Usually comes before detailed requirements digging

Requirements+problem prerequisite

- What is it and how does it help?



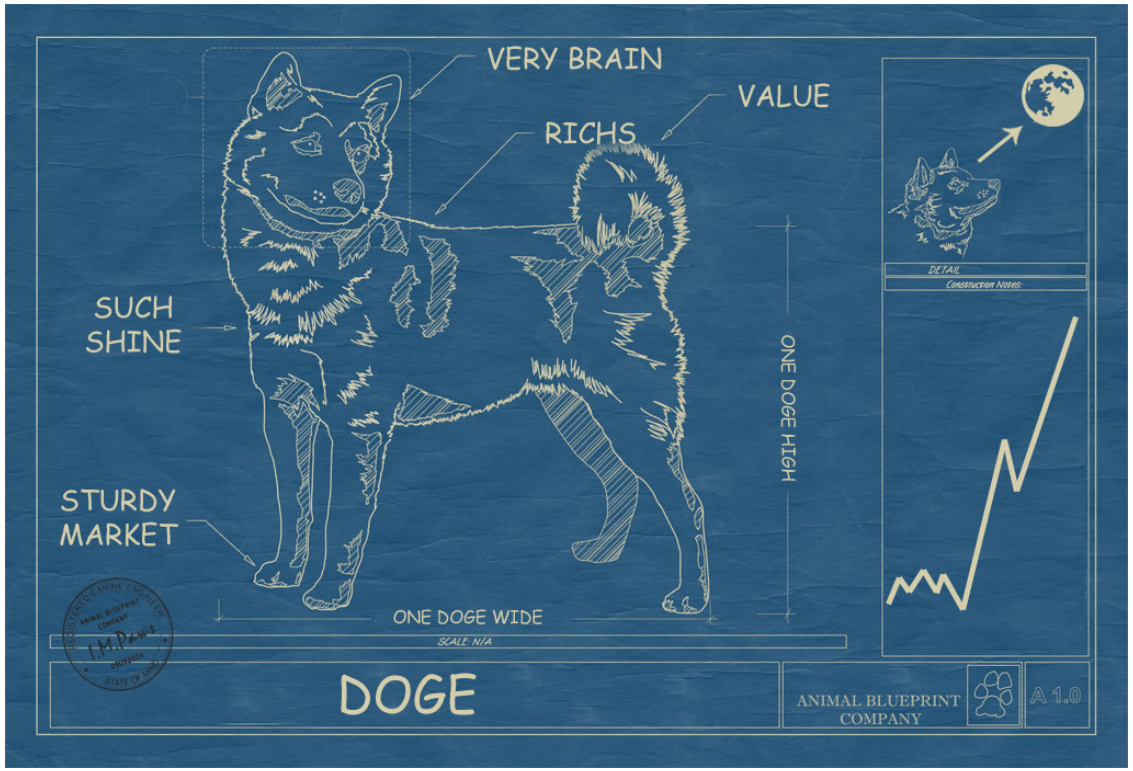
Picturing architecture = communication

- From Allen:
  - Can’t code effectively unless you can talk to other folks about what you’re doing
  - Again, all that we do is a social effort

- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.

Picturing architecture = communication

- From Allen:
  - Can’t code effectively unless you can talk to other folks about what you’re doing
  - Again, all that we do is a social effort
- Building a visual toolkit to visualize the structure and behavior of code
  - Using pictures, index cards, matrix charts, diagrams, UML, etc.



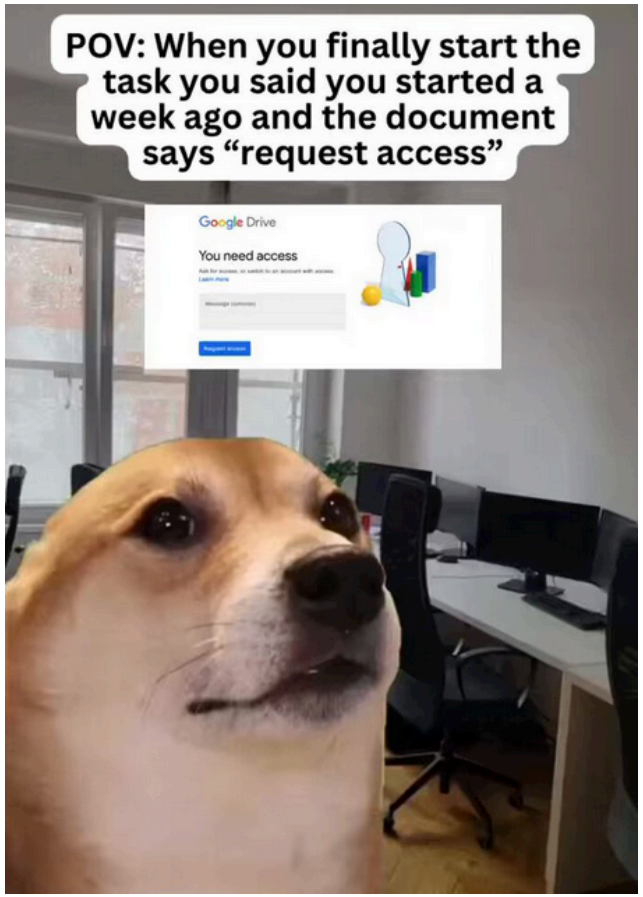
- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.

Architecture prerequisites

- Also called system architecture, high-level design, top-level design
- Program organization & all interconnected relationships
  - Context
  - User roles and permissions
  - Data flow
  - Classes
  - Business rules
  - Graphical user interface (GUI)
  - Resource management & performance (memory, GPU)
  - Security
  - Interoperability (does the program share resources?)
  - Scalability (can the system evolve to future demands?)
  - Feasibility (concerns/risks)
  - Buy vs. build decisions (pros and cons)
  - Change management strategy (operational and organizational)
  - Etc…

Defining access control

- Stemming from the problem and requirement prerequisites



- Escape (2024). Understanding Access Control Models. Online: https://escape.tech/blog/access-control-models/

Defining access control

- Stemming from the problem and requirement prerequisites
- Models
  - RBAC
  - ABAC
  - DAC

Allowable Action	Admin	Manager	User	Guest
Create Records	✓	✓	✗	✗
Read/View Records	✓	✓	✓	✓
Update/Edit Records	✓	✓	✓	✗
Delete Records	✓	✗	✗	✗
Manage Permissions	✓	✗	✗	✗

- Escape (2024). Understanding Access Control Models. Online: https://escape.tech/blog/access-control-models/

Defining access control

- Stemming from the problem and requirement prerequisites
- Models
  - RBAC
  - ABAC
  - DAC

Allowable Action	Admin Condition	Manager-in-Dept Condition	Record Owner Condition
Create Records	`isAdmin = true` → ✅	—	—
Read/View Records	`isAdmin = true` → ✅	`jobTitle = "Manager"` **AND** `user.department = resource.department` → ✅	`userId = resource.ownerId` → ✅
Update/Edit Records	`isAdmin = true` → ✅	`jobTitle = "Manager"` **AND** `user.department = resource.department` → ✅	—
Delete Records	`isAdmin = true` → ✅	—	—
Manage Permissions	`isAdmin = true` → ✅	—	—

- Escape (2024). Understanding Access Control Models. Online: https://escape.tech/blog/access-control-models/

Defining access control

- Stemming from the problem and requirement
- Models
  - RBAC
  - ABAC
  - DAC

Allowable Action	Admin	Manager (same dept as record)	Record Owner (Employee)
Create Records	✓ *(by policy/ administrative privilege)*	✗	✗
Read/View Records	✓ *(ACL or global admin)*	✓ *(if owner added dept manager to ACL)*	✓ *(owner of the record)*
Update/Edit Records	✓	✓ *(if owner added dept manager to ACL)*	✗ *(owner retains read only in the given scenario)*
Delete Records	✓	✗	✗
Manage Permissions (ACL)	✓ *(admin override)*	✗ *(unless owner delegates)*	✓ *(as the resource owner)*

- Escape (2024). Understanding Access Control Models. Online: https://escape.tech/blog/access-control-models/

Defining access control

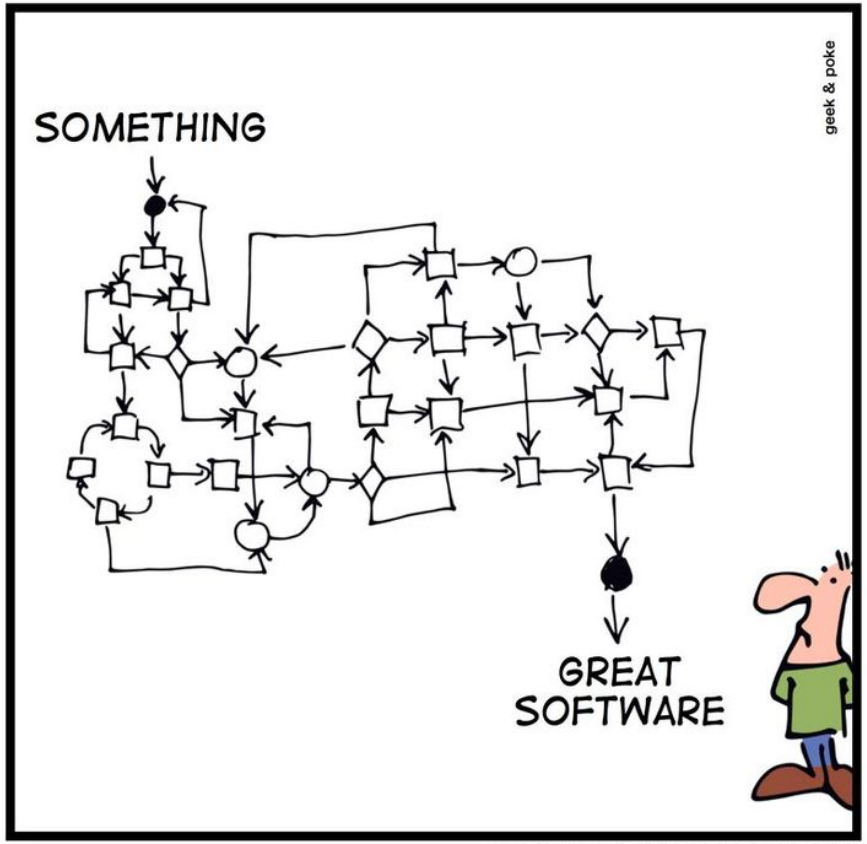
- Stemming from the problem and requirement prerequisites
- Models
  - RBAC
  - ABAC
  - DAC
- Which to use?

Model	Basis	Strengths	Weaknesses
RBAC	Roles	Simple, scalable, widely implemented	Rigid when environments change
ABAC	Attributes	Highly flexible and granular	Complex to design and maintain
DAC	Ownership	Owners retain full control	Weaker security; prone to propagation errors

- Escape (2024). Understanding Access Control Models. Online: https://escape.tech/blog/access-control-models/

Unified Modelling Language (UML)

- Standard modelling/pictorial language

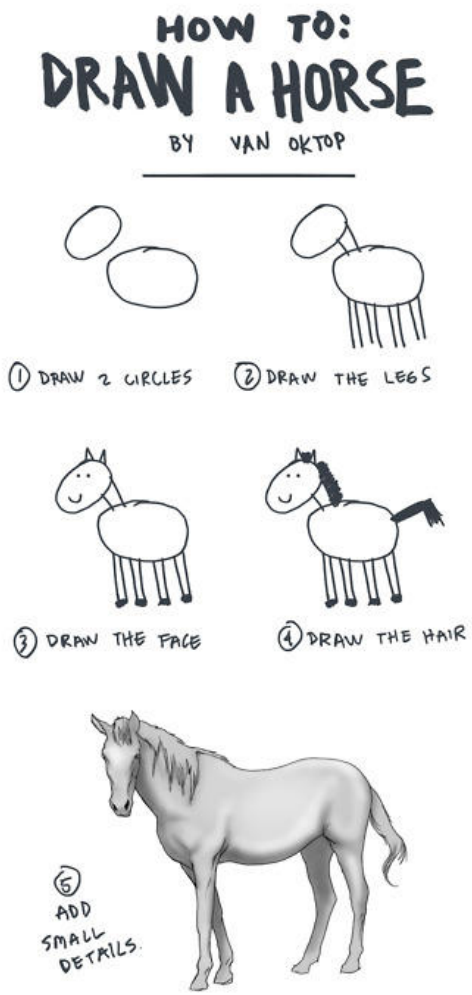


DEVELOPMENT PROCESS

- TutorialsPoint. (Online). https://www.tutorialspoint.com/uml/
- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.

Unified Modelling Language (UML)

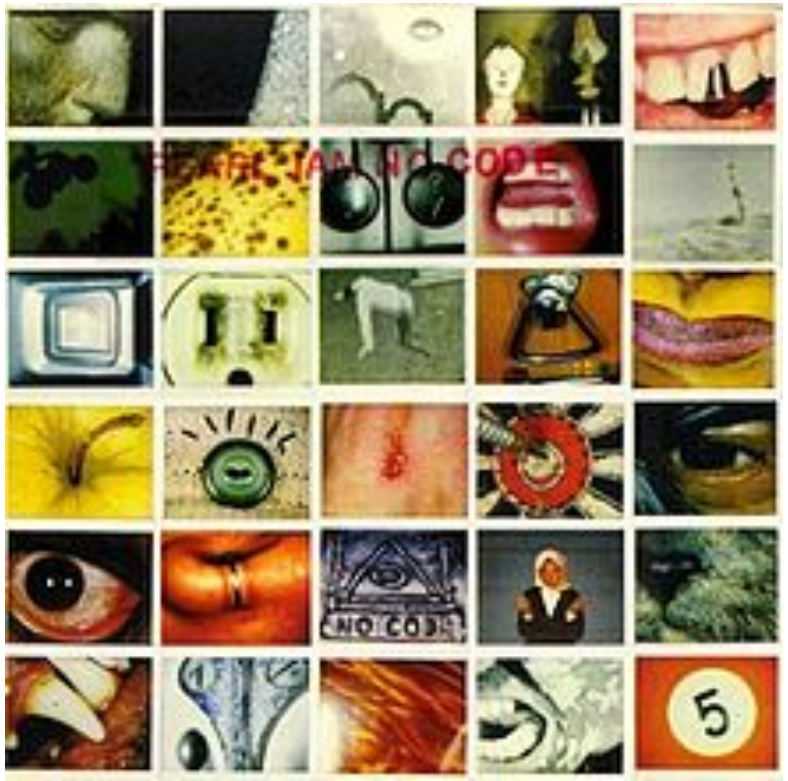
- Standard modelling/pictorial language
- “Picture is worth 1000 words”
  - Everyone/all skill levels should be able to understand it



- TutorialsPoint. (Online). https://www.tutorialspoint.com/uml/
- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.

Unified Modelling Language (UML)

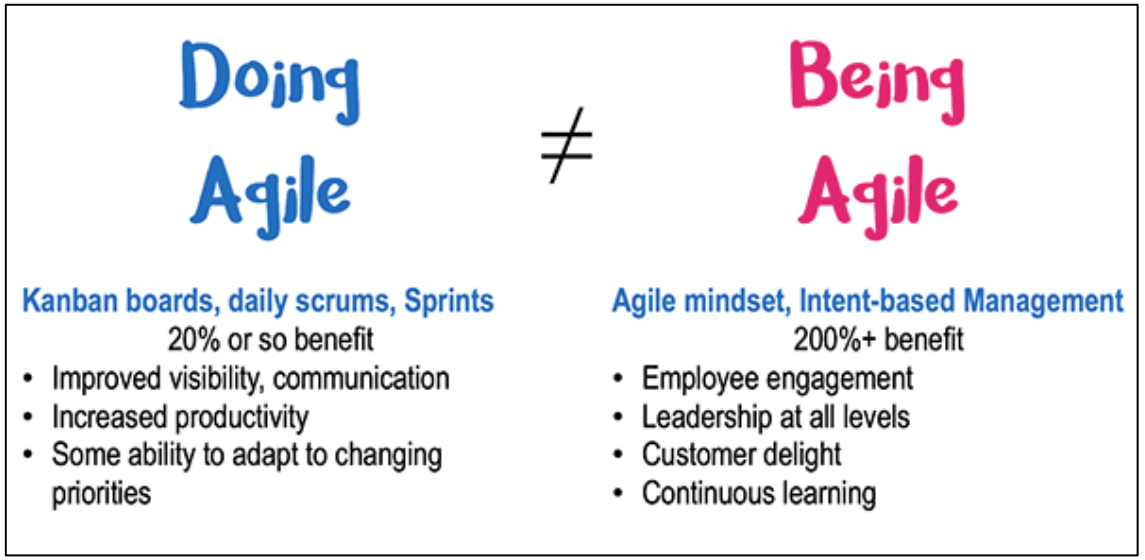
- Standard modelling/pictorial language
- “Picture is worth 1000 words”
  - Everyone/all skill levels should be able to understand it
- No coding required
  - Tools exist to generate code based on diagramming, and vice versa
    - Be warned – Visuals can get complex too though & can get too close to code



- TutorialsPoint. (Online). https://www.tutorialspoint.com/uml/
- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.

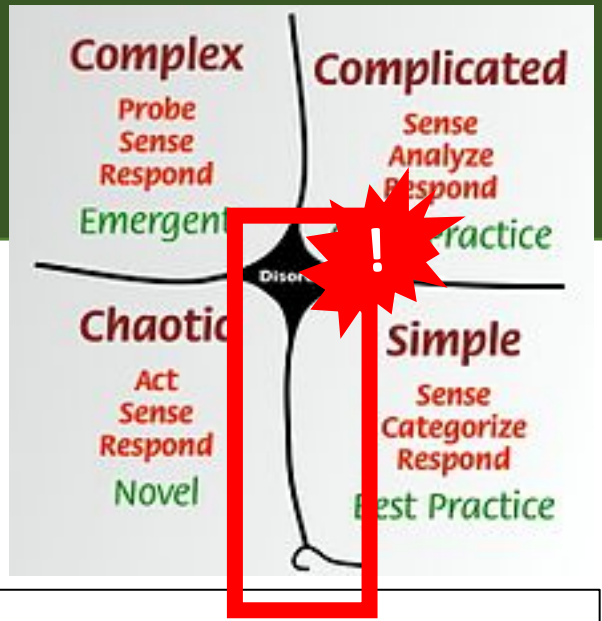
Unified Modelling Language (UML)

- Standard modelling/pictorial language
- "Picture is worth 1000 words"
  - Everyone/all skill levels should be able to understand it
- No coding required
  - Tools exist to generate code based on diagramming, and vice versa
    - Be warned – Visuals can get complex too though & can get too close to code
- Note on "doing vs. being agile"

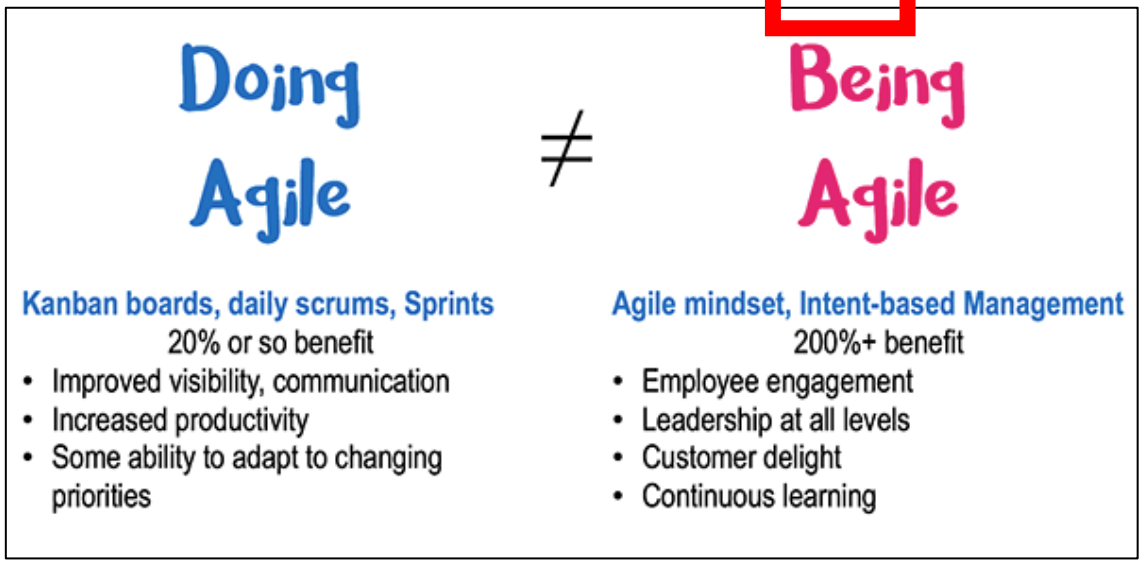


- TutorialsPoint. (Online). https://www.tutorialspoint.com/uml/
- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.

Unified Modelling Language (UML)



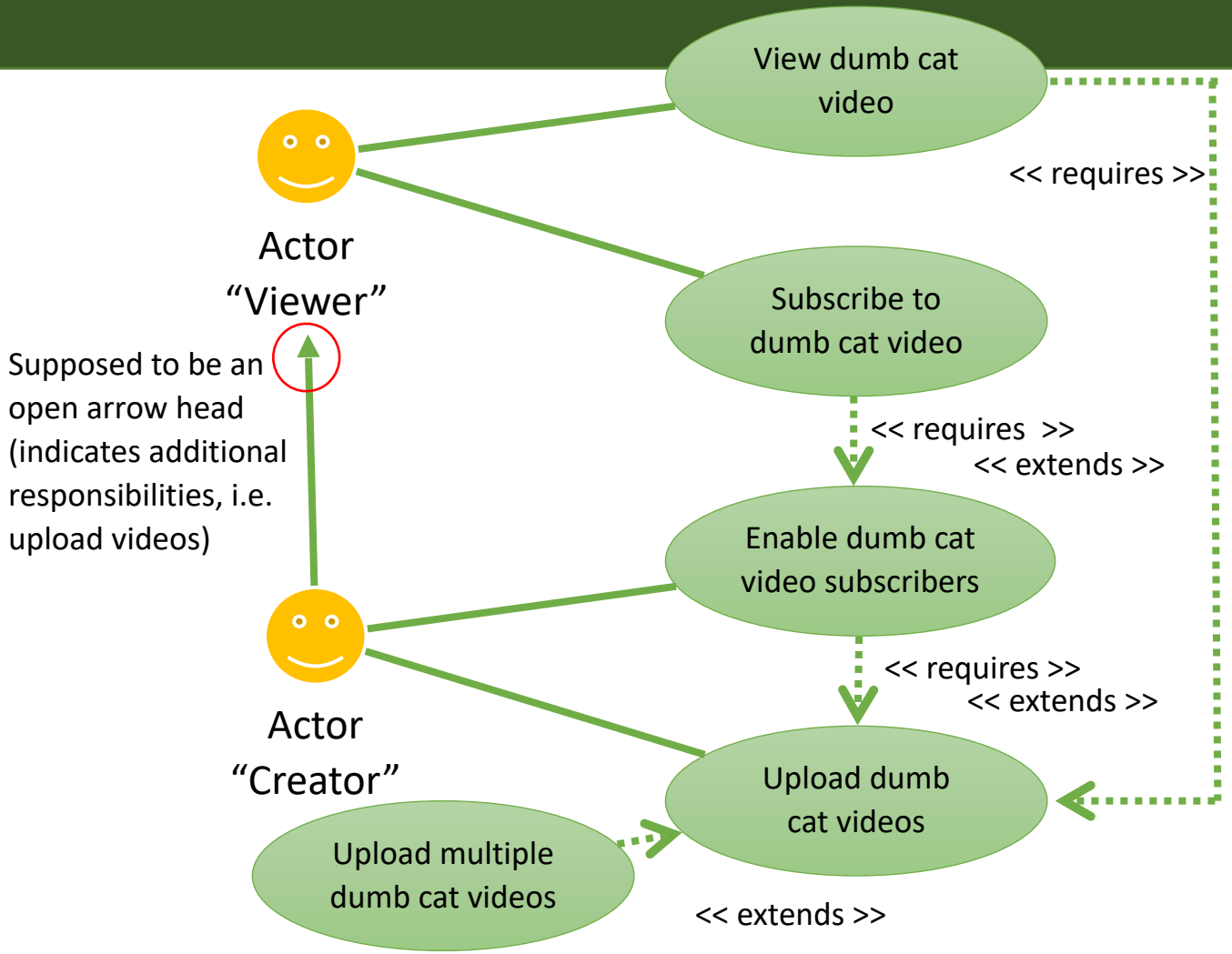
- Standard modelling/pictorial language
- "Picture is worth 1000 words"
  - Everyone/all skill levels should be able to understand it
- No coding required
  - Tools exist to generate code based on diagramming, and vice versa
    - Be warned – Visuals can get complex too though & can get too close to code
- Note on "doing vs. being agile"



- TutorialsPoint. (Online). https://www.tutorialspoint.com/uml/
- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.

Use case (story) diagramming

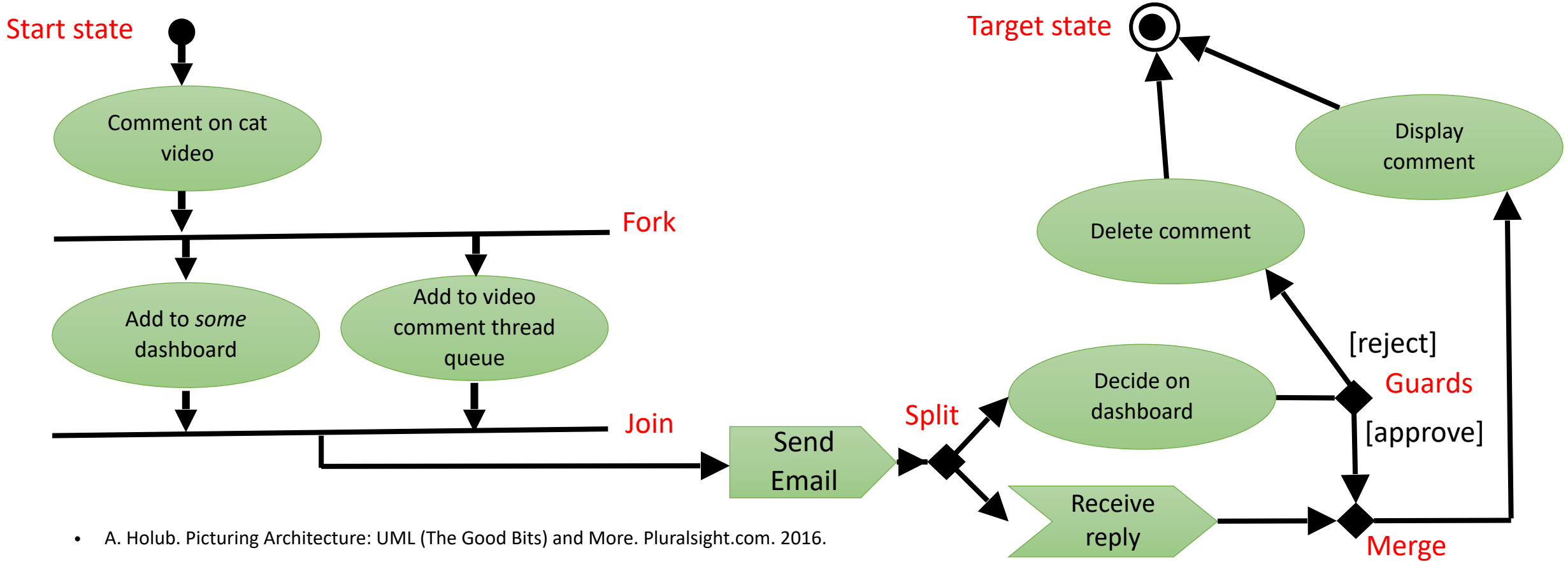
- Represent user (type) activities and their dependencies and relationships
- Discussion
  - Would your parents understand this pic?
  - Can get complex



- A. Holub. Allen Holub's UML Quick Reference. Online: http://holub.com/uml/
- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.

Activity (state) diagramming

- Represent specific activities and their dependencies and relationships (doesn't matter who's doing the activity)



- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.

Interaction diagramming

- How objects interact at run time to do the *stuff and things* as per Stakeholder needs
- Focuses on the messages sent and received between objects
- Typically a single use case/interaction
- Two types of diagrams
  - Communication/collaboration
  - Sequence

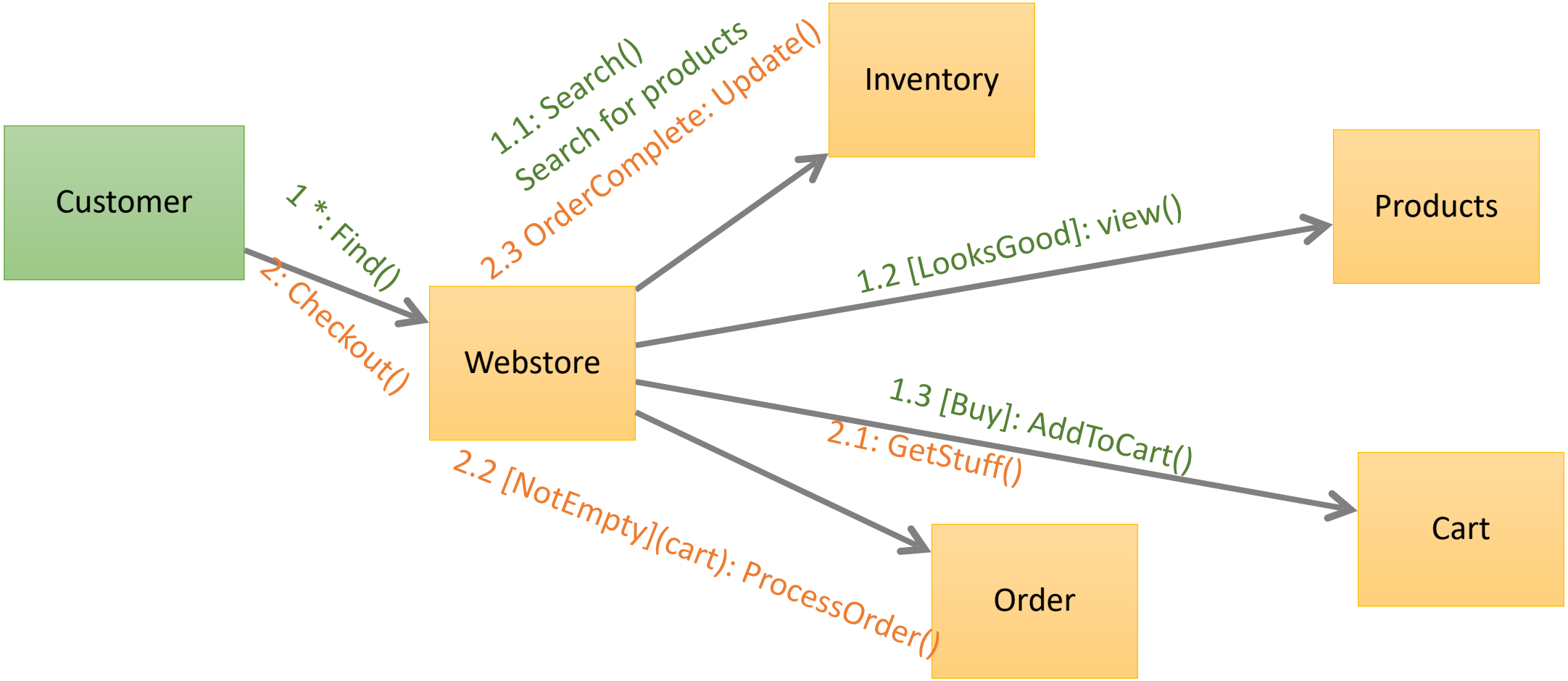
- A. Wick. Requirements Elicitation and Analysis. Lynda.com. 2017.
- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016

Communication/collaboration diagrams

- Visualization of how objects interact with each other
- Objects in boxes
- Numbers and arrows represent communication order and flow
- "*" means a message is sent more than once
- "order#[condition]: some msg" means a message is sent if the condition = true

- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.

E.g. Communication/collaboration diagrams



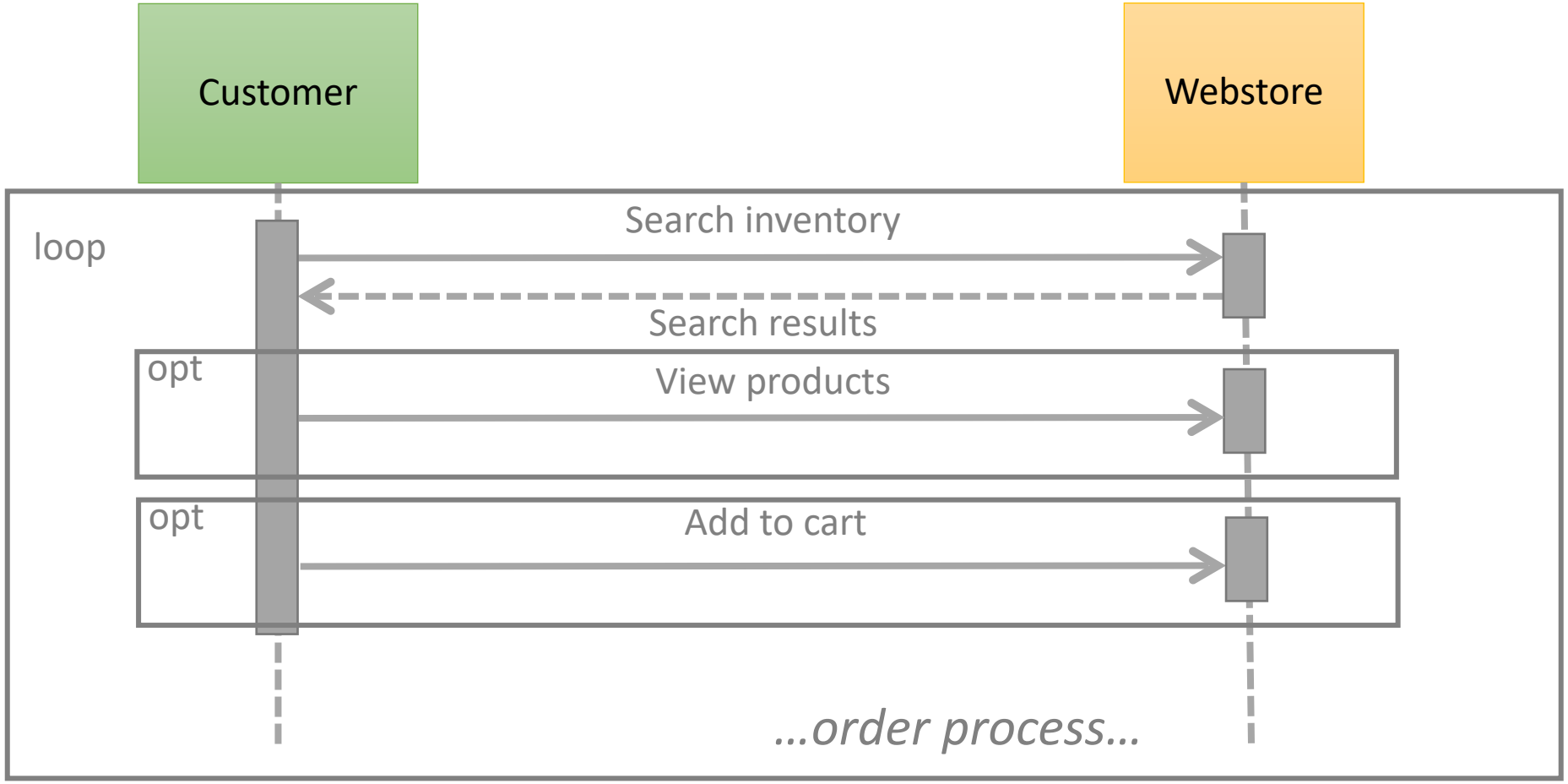
- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.
- UML Diagram Examples, Online Shopping. Online: https://www.uml-diagrams.org/examples/online-shopping-example.html

Sequence diagrams

- Sequence diagram, not a sequin diagram ☺
- Again, visualization of how objects interact with each other
- Messages are on the horizontal
- Objects/Time represented on the vertical
  - Represented as a dotted line stemming from object(s) – “lifeline”
- Boxes covering lifelines are called activations
  - Length of boxes represent time
- Whole bunch of other stuff…
  - Loops, timeouts, optionals, etc.
  - Holub: *Getting close to code, why not just code?!?*

- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.

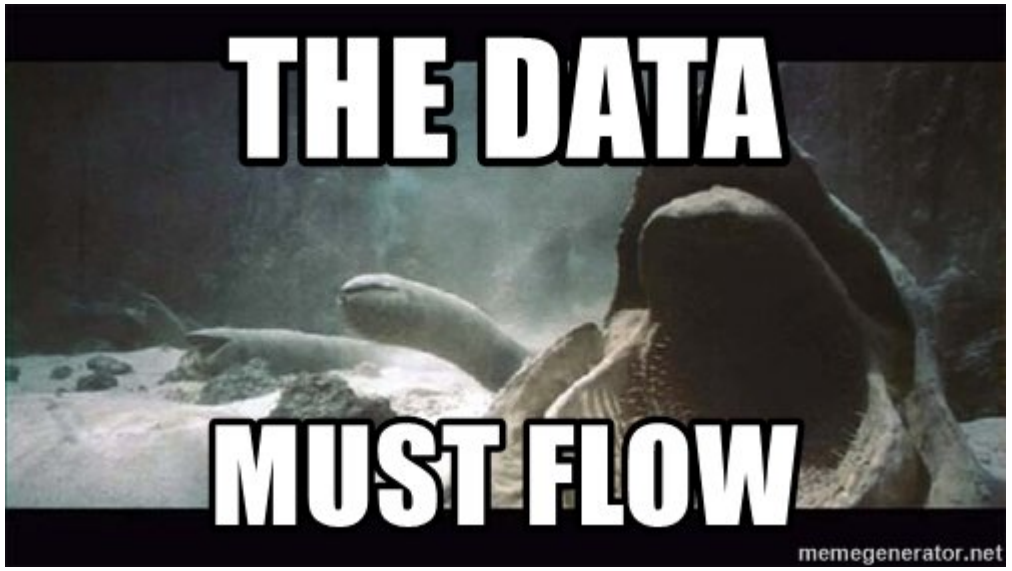
E.g. Sequence diagrams



- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.
- UML Diagram Examples, Online Shopping. Online: https://www.uml-diagrams.org/online-shopping-uml-sequence-diagram-example.html

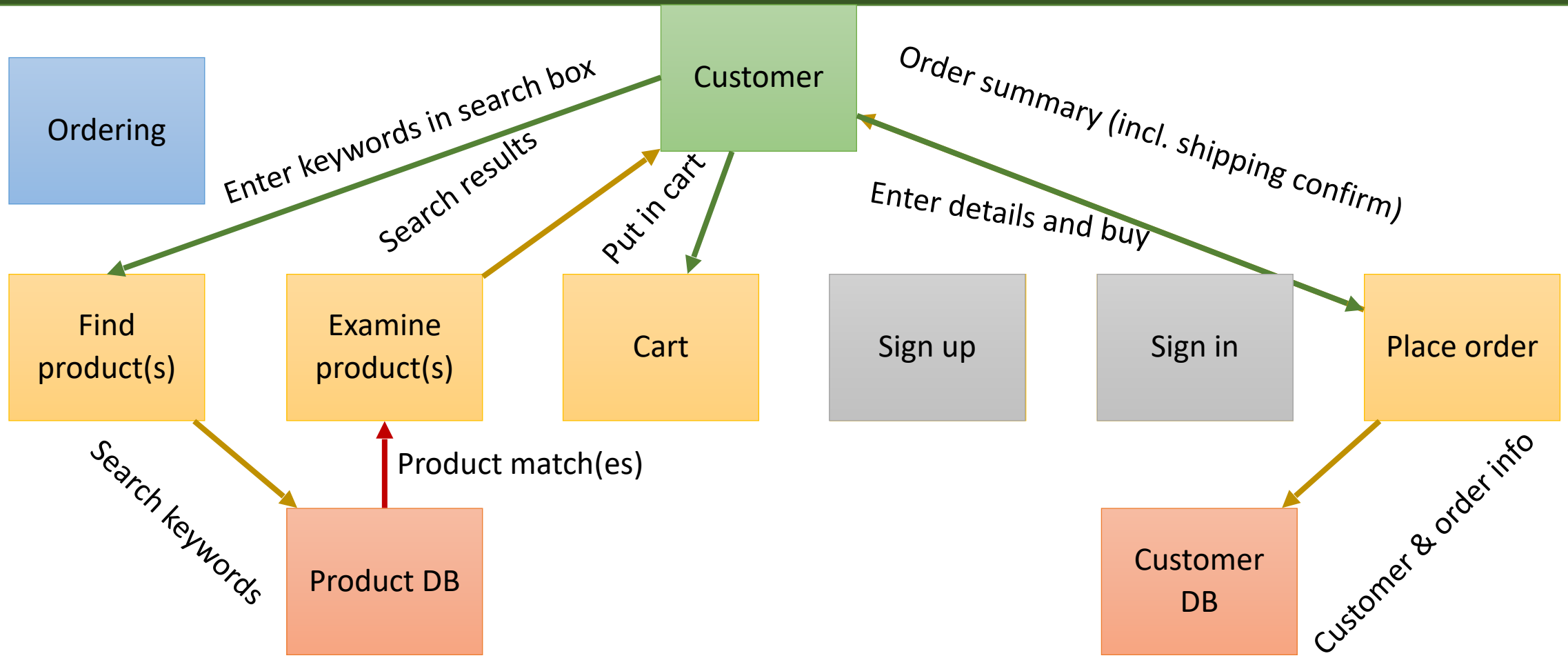
Data (flow) diagramming

- Mapping/visualizing the processes and associated data flow
  - Why?
  - Can get a better understanding of the system by examining the data flows
- Collaboration with Stakeholders (customers/clients, etc.) and development team
  - Eliciting requirements
  - Walking through all of the processes with a focus on data needs



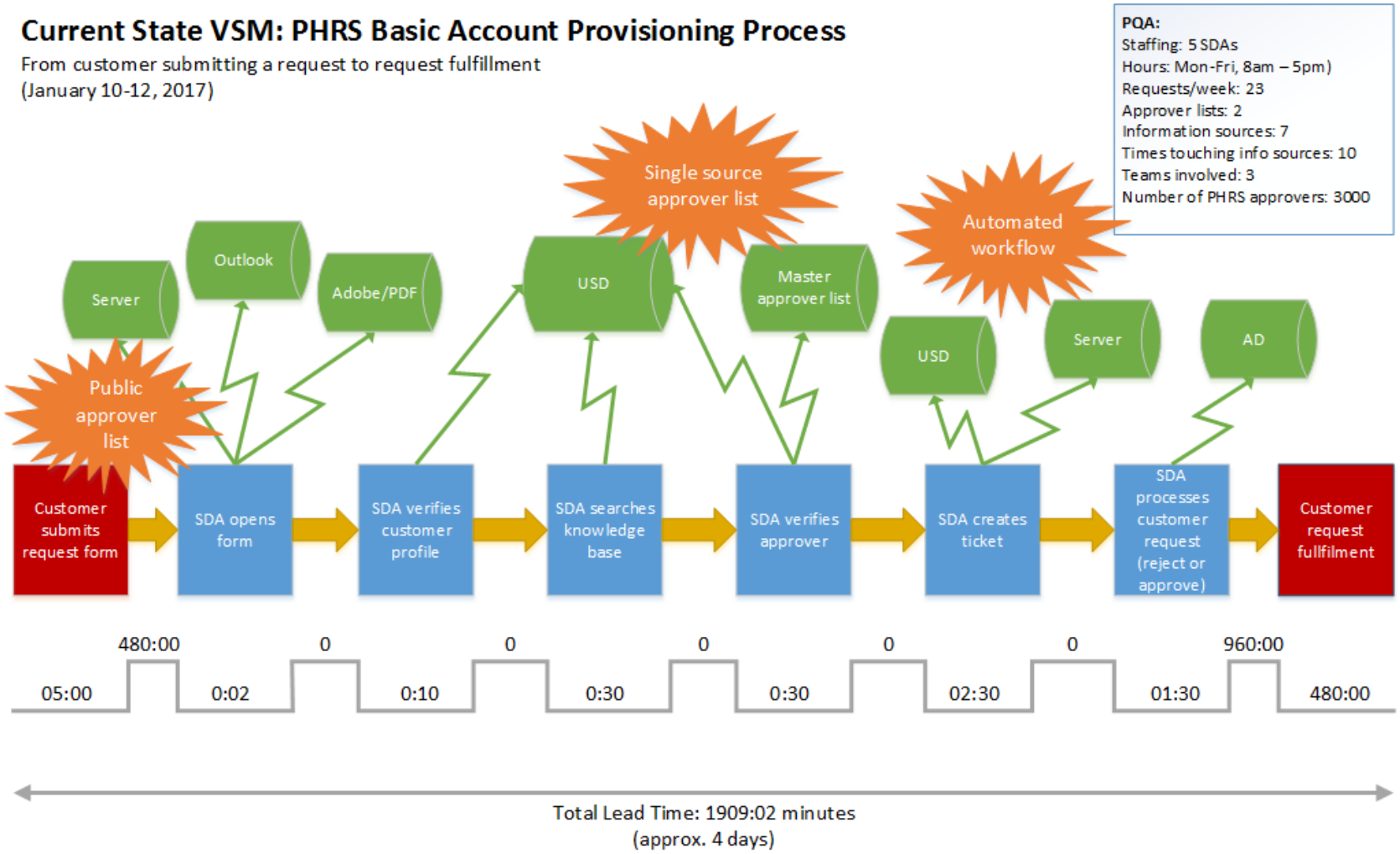
- A. Wick. Requirements Elicitation and Analysis. Lynda.com. 2017.

E.g. Data flow diagramming

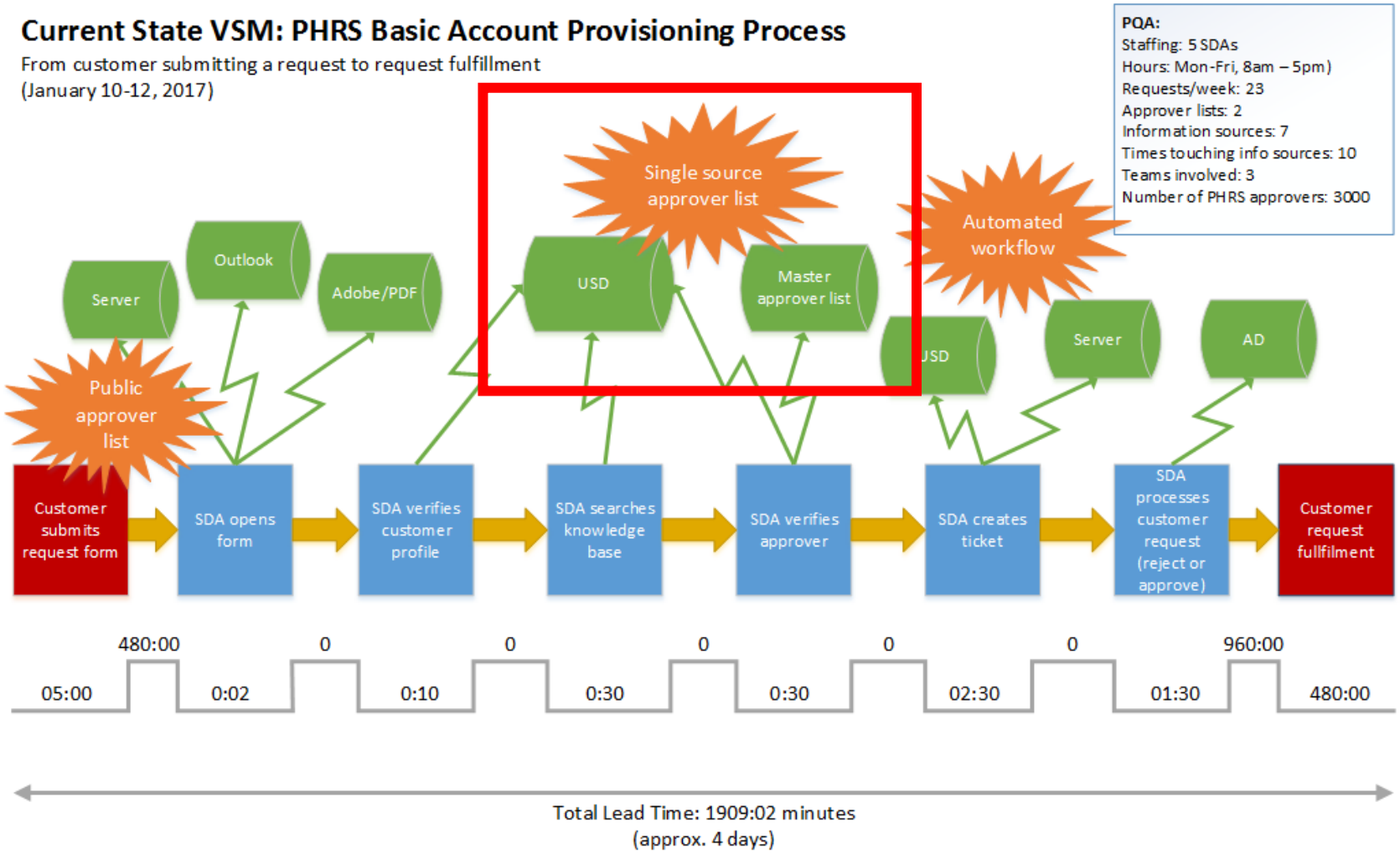


- A. Wick. Requirements Elicitation and Analysis. Lynda.com. 2017.
- StoriesOnBoard (Image). Online: http://storiesonboard.com/

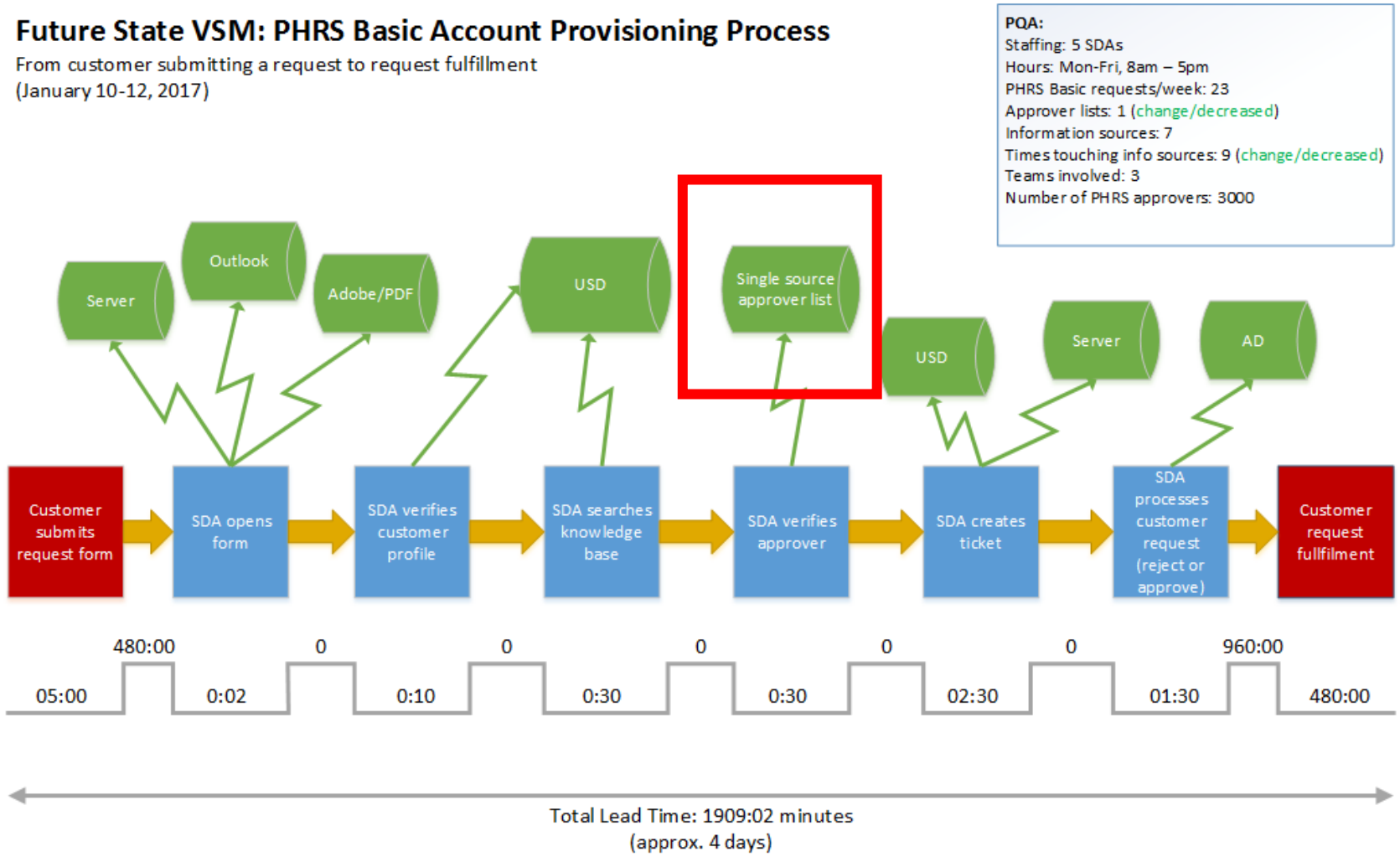
Lean (Current/Future) Value Stream Mapping



Lean (Current/Future) Value Stream Mapping

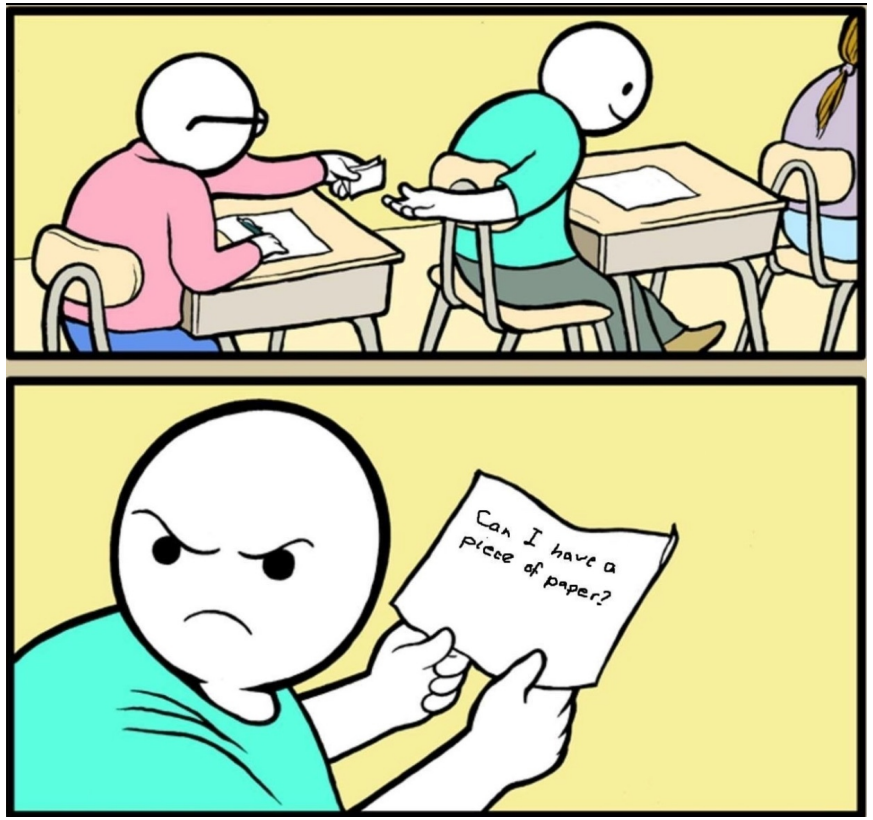


Lean (Current/Future) Value Stream Mapping



Class diagrams

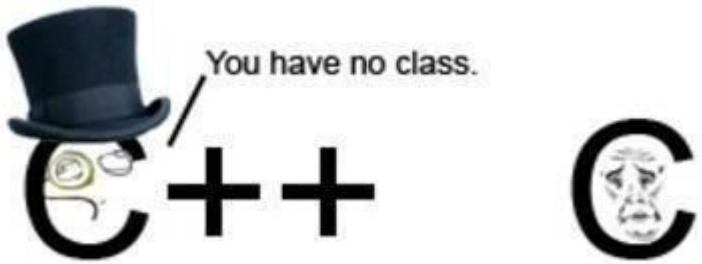
- Important note



- A. Holub. UML Quick Reference (2017). Online: http://holub.com/uml/
- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.

Class diagrams

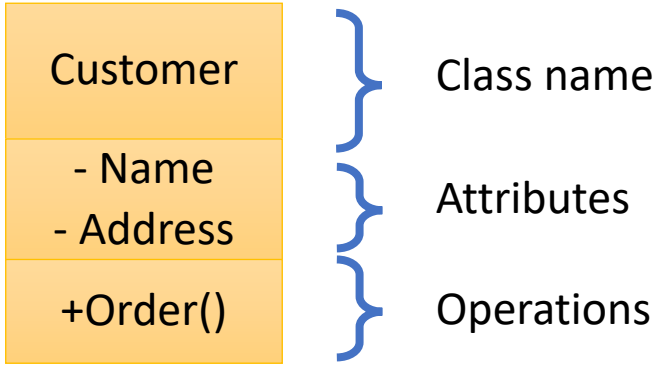
- Important note
- Visualize the relationship between classes and objects



- A. Holub. UML Quick Reference (2017). Online: http://holub.com/uml/
- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.

Class diagrams

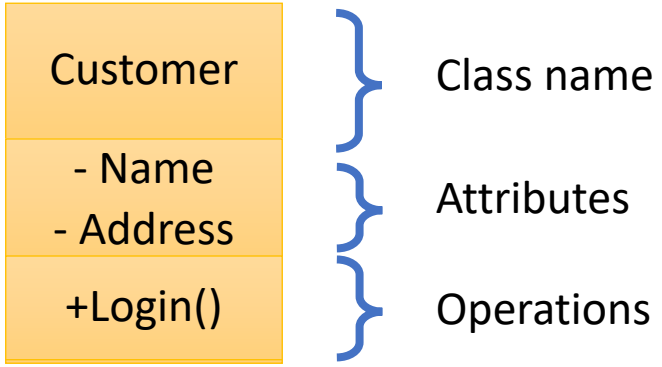
- Important note
- Visualize the relationship between classes and objects
- Representation



- A. Holub. UML Quick Reference (2017). Online: http://holub.com/uml/
- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.

Class diagrams

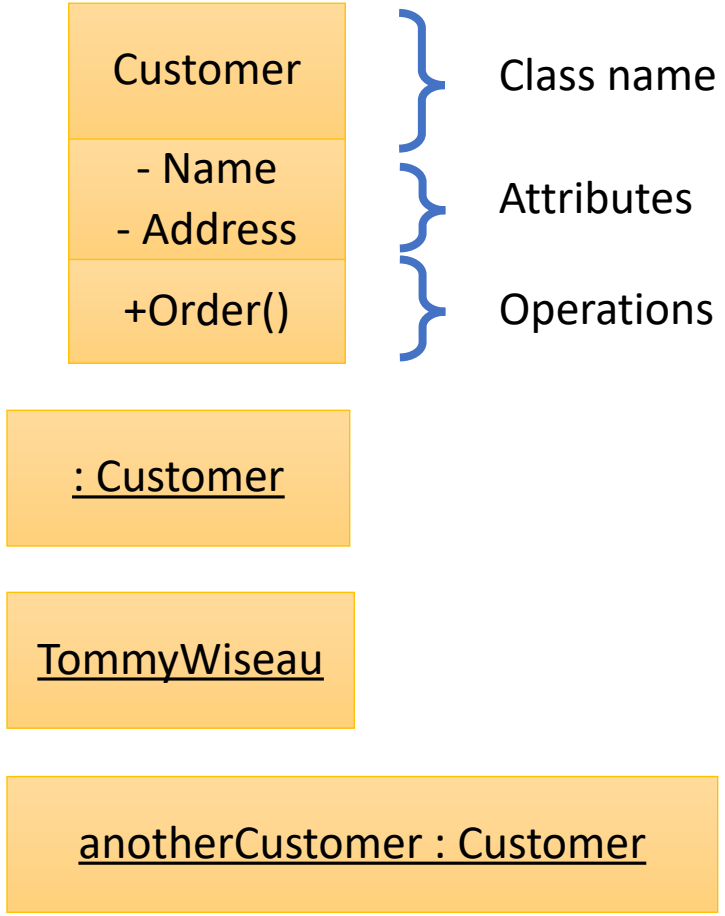
- Important note
- Visualize the relationship between classes and objects
- Representation



- A. Holub. UML Quick Reference (2017). Online: http://holub.com/uml/
- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.

Class diagrams

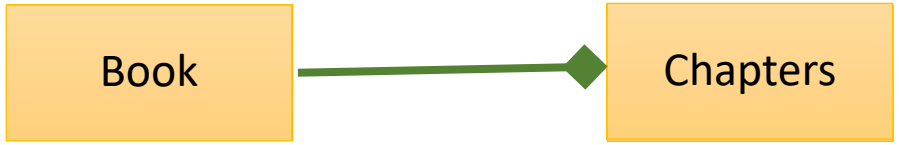
- Important note
- Visualize the relationship between classes and objects
- Representation



- A. Holub. UML Quick Reference (2017). Online: http://holub.com/uml/
- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.

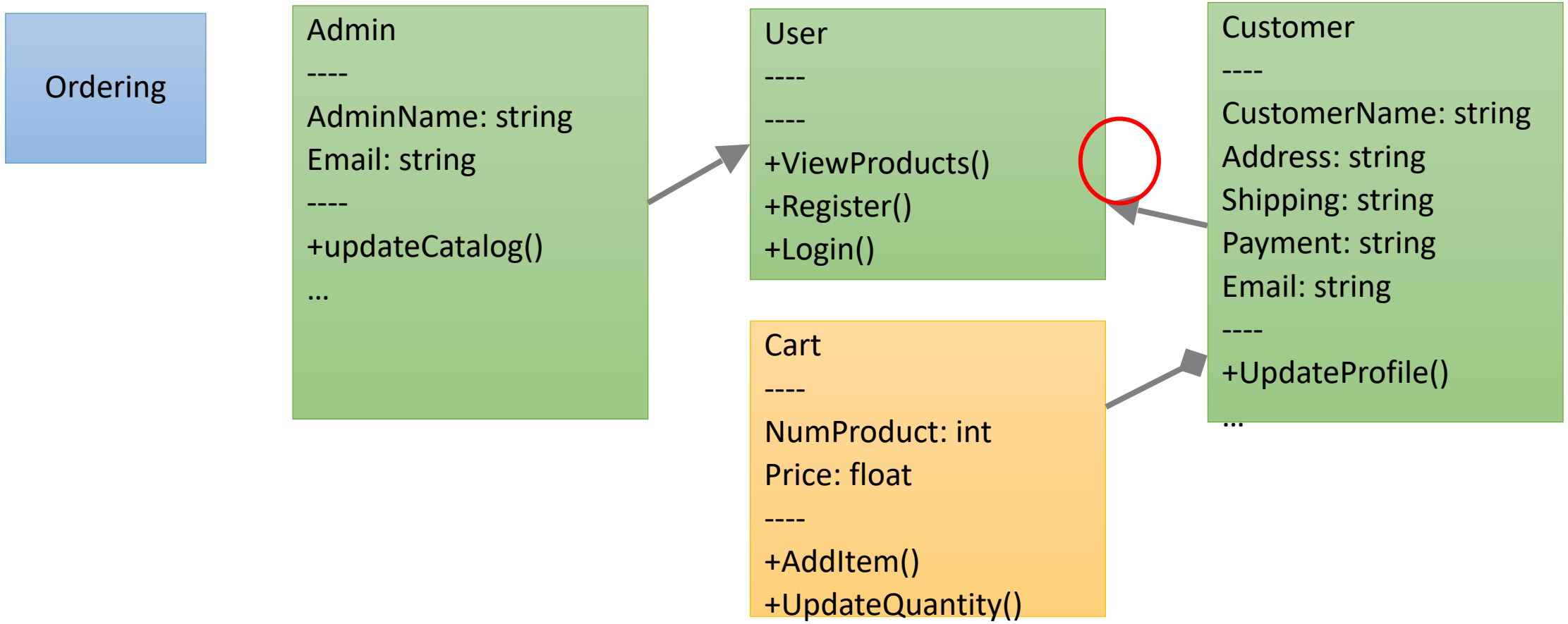
Class diagrams

- Important note
- Visualize the relationship between classes and objects
- Representation
- A lot of weird arrows
  - Inheritance/Generalization
  - Aggregation hierarchy
  - Composition hierarchy



- A. Holub. UML Quick Reference (2017). Online: http://holub.com/uml/
- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.

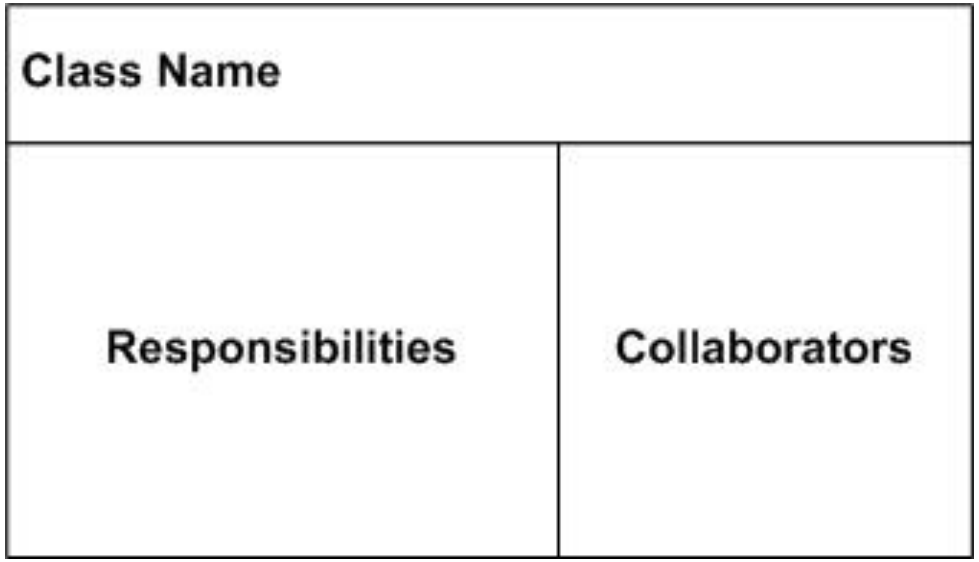
E.g. Class diagram for online shopping



- LucidChart. Class Diagram for Online Shopping System. Online: https://www.lucidchart.com/documents/edit/4c127ee0-7652-4bb9-b53c-738e2c3a4a35?demo=on#

CRC cards

- What are these?



- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.
- O. Elgabry. Object-Oriented Analysis and Design: Conceptual Model (Part 2) (2017). Online: https://medium.com/omarelgabrys-blog/object-oriented-analysis-and-design-conceptual-model-part-2-ce730ac4eb31

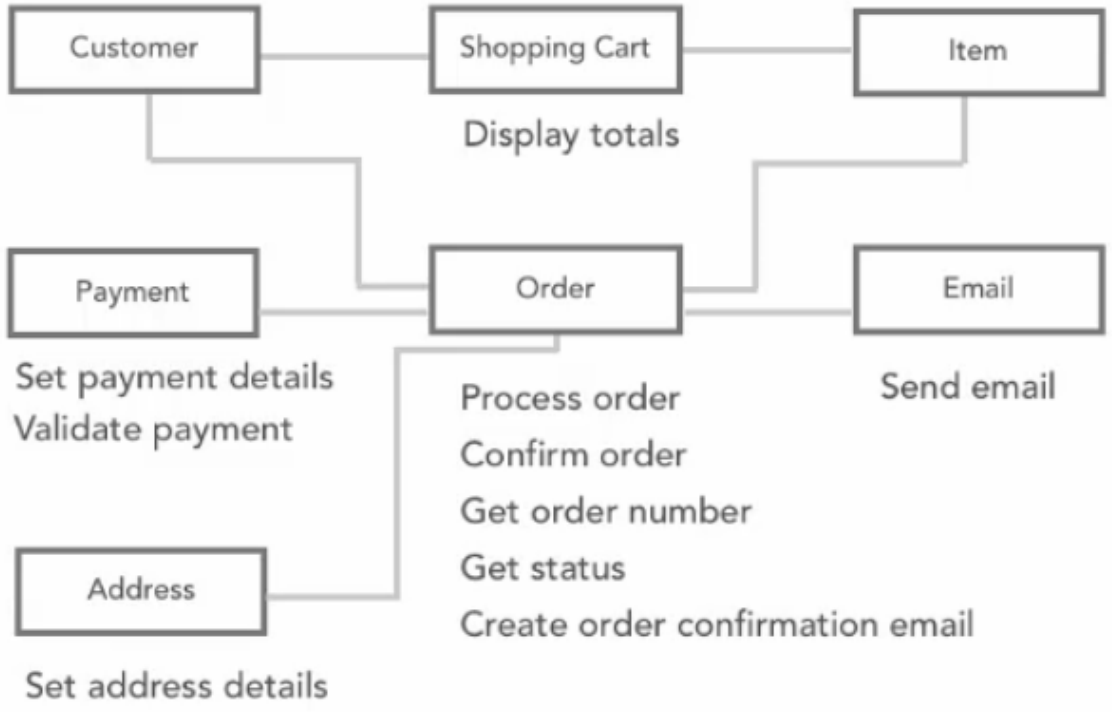
CRC cards

- What are these?
- Visual
  - Index cards representing classes
  - Relationships between objects
  - Responsibilities of the class

- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.
- O. Elgabry. Object-Oriented Analysis and Design: Conceptual Model (Part 2) (2017). Online: https://medium.com/omarelgabrys-blog/object-oriented-analysis-and-design-conceptual-model-part-2-ce730ac4eb31

CRC cards

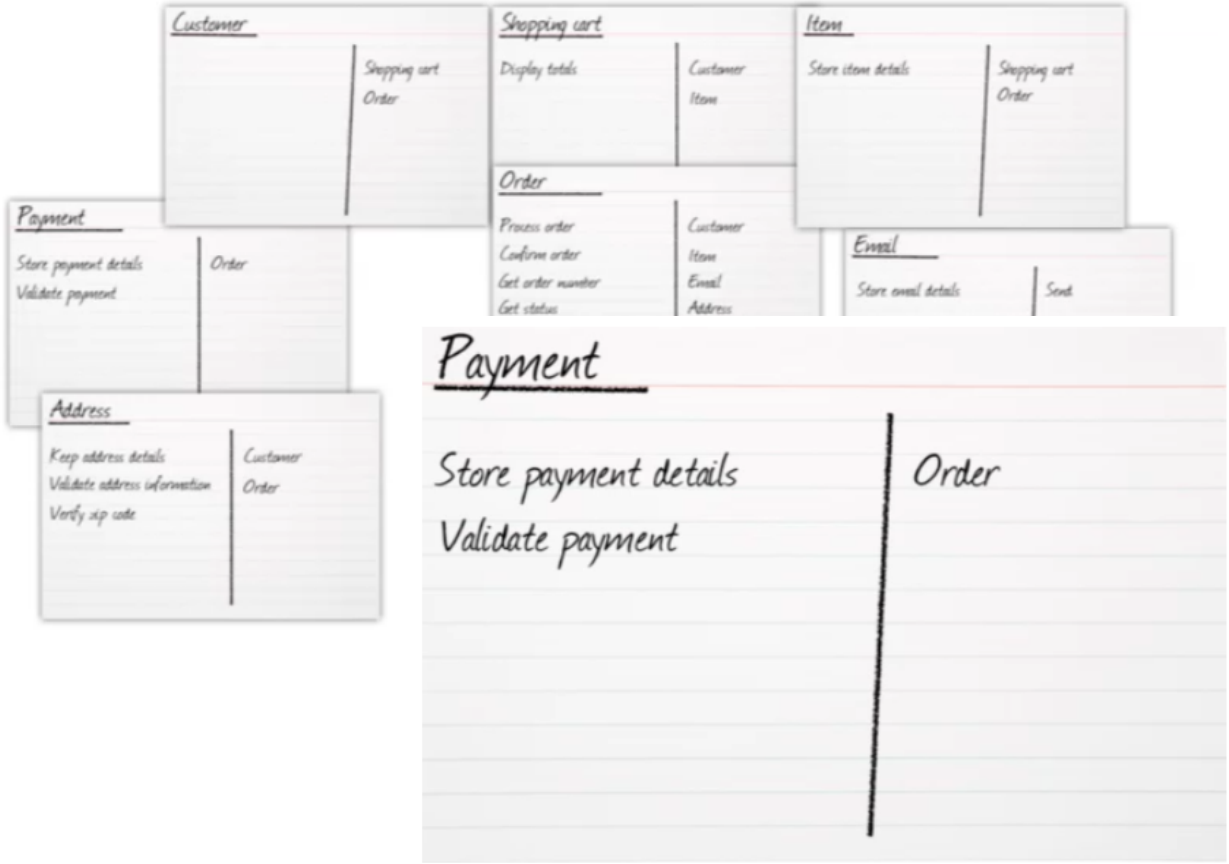
- What are these?
- Visual
  - Index cards representing classes
  - Relationships between objects
  - Responsibilities of the class
- Notes



- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.
- O. Elgabry. Object-Oriented Analysis and Design: Conceptual Model (Part 2) (2017). Online: https://medium.com/omarelgabrys-blog/object-oriented-analysis-and-design-conceptual-model-part-2-ce730ac4eb31

CRC cards

- What are these?
- Visual
  - Index cards representing classes
  - Relationships between objects
  - Responsibilities of the class
- Notes



- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight.com. 2016.
- O. Elgabry. Object-Oriented Analysis and Design: Conceptual Model (Part 2) (2017). Online: https://medium.com/omarelgabrys-blog/object-oriented-analysis-and-design-conceptual-model-part-2-ce730ac4eb31

A great resource

- A. Holub. Allen Holub's UML Quick Reference (2017). Online: https://holub.com/uml/

Picturing design: Lo-Fidelity (Lofi) prototyping

- You are not the user
  - ENSE 271 experience…
  - Recall: False-Consensus effect
- Lo-fi prototyping
- Quick way to
  - Realize system vision (design)
  - Test your design (the interface of your app is the user's doorway

- Nielsen Norman Group. The False-Consensus Effect (2018). Online: https://www.nngroup.com/videos/false-consensus-effect/
- M. Wettergreen. Low Fidelity Prototyping Example (2015). Online: https://youtu.be/1LRuQgEZTU4

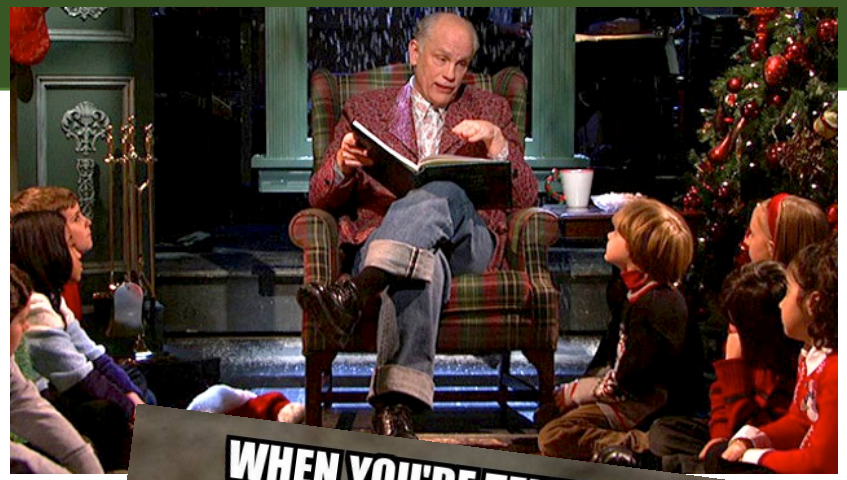
Domain driven design & user stories

- About Domain Driven Design
  - E.g. Accounting system

- Allen Holub. Software Architecture Foundations. LinkedIn Learning. 2019. Online: https://www.linkedin.com/learning/software-architecture-foundations
- T.A. Alspaugh. Narrative Requirements. 2015. Online: http://www.thomasalspaugh.org/pub/fnd/narrativeRequirements.html
- M. Fowler. Bounded Context (2014). Online: https://martinfowler.com/bliki/BoundedContext.html

Domain driven design & user stories

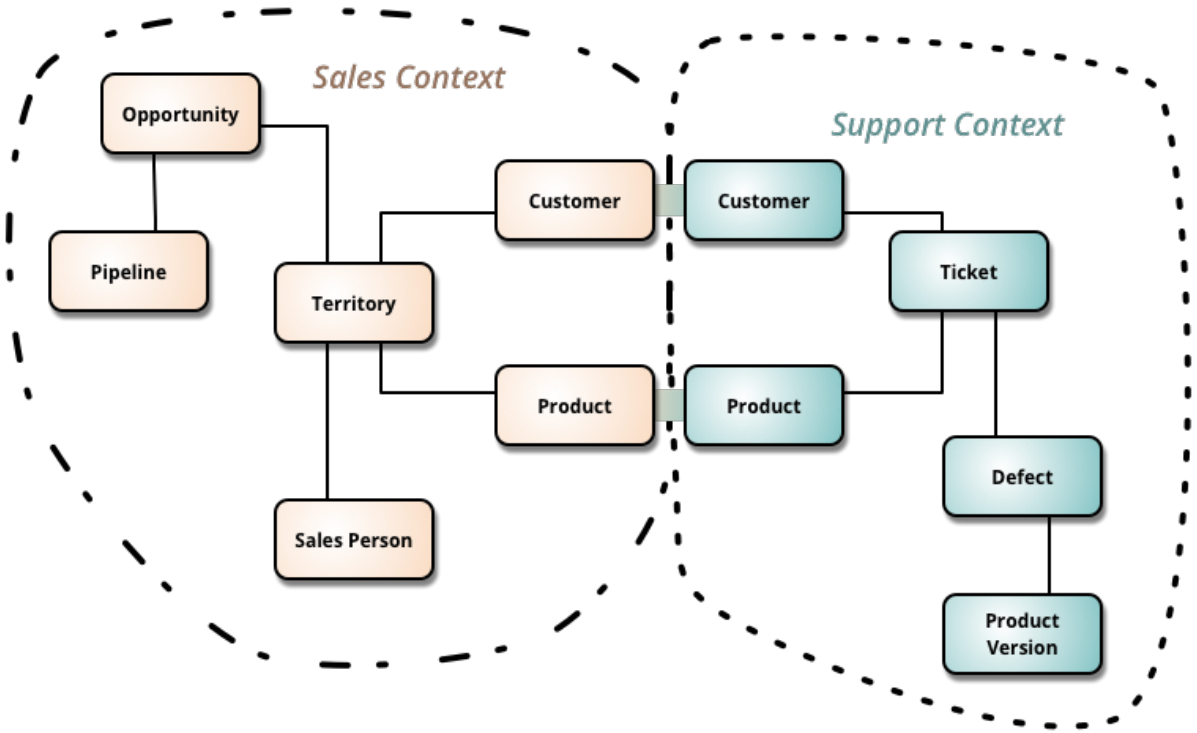
- About Domain Driven Design
  - E.g. Accounting system
- Visualization through user stories



- Allen Holub. Software Architecture Foundations. LinkedIn Learning. 2019. Online: https://www.linkedin.com/learning/software-architecture-foundations
- T.A. Alspaugh. Narrative Requirements. 2015. Online: http://www.thomasalspaugh.org/pub/fnd/narrativeRequirements.html
- M. Fowler. Bounded Context (2014). Online: https://martinfowler.com/bliki/BoundedContext.html

Domain driven design & user stories

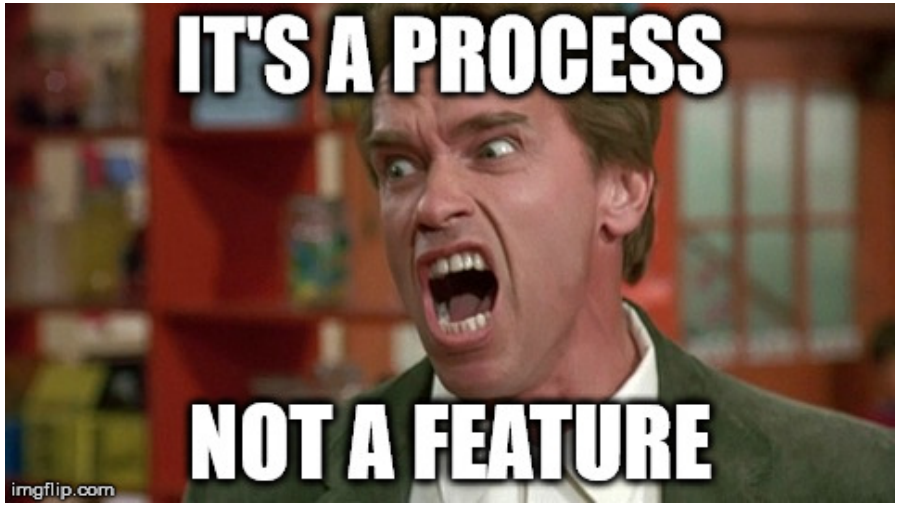
- About Domain Driven Design
  - E.g. Accounting system
- Visualization through user stories
- Supports “bounded context”



- Allen Holub. Software Architecture Foundations. LinkedIn Learning. 2019. Online: https://www.linkedin.com/learning/software-architecture-foundations
- T.A. Alspaugh. Narrative Requirements. 2015. Online: http://www.thomasalspaugh.org/pub/fnd/narrativeRequirements.html
- M. Fowler. Bounded Context (2014). Online: https://martinfowler.com/bliki/BoundedContext.html

User story specifics

- About user stories
- Holub describes a story as “a narrative that takes a user through some process that results in a valuable outcome”



- A. Holub. Picturing Architecture: UML (The Good Bits) and More. Pluralsight. 2016.
- Merriam-Webster. Story. Online: https://www.merriam-webster.com/dictionary/story

Visualizing work

- Kanban board
  - Product backlog
    - To do/ ice box
    - Comprised of user stories
  - In-progress
  - Done

- Image from HBO's Silicon Valley - What's the emergency Erlich!!?? (haha)

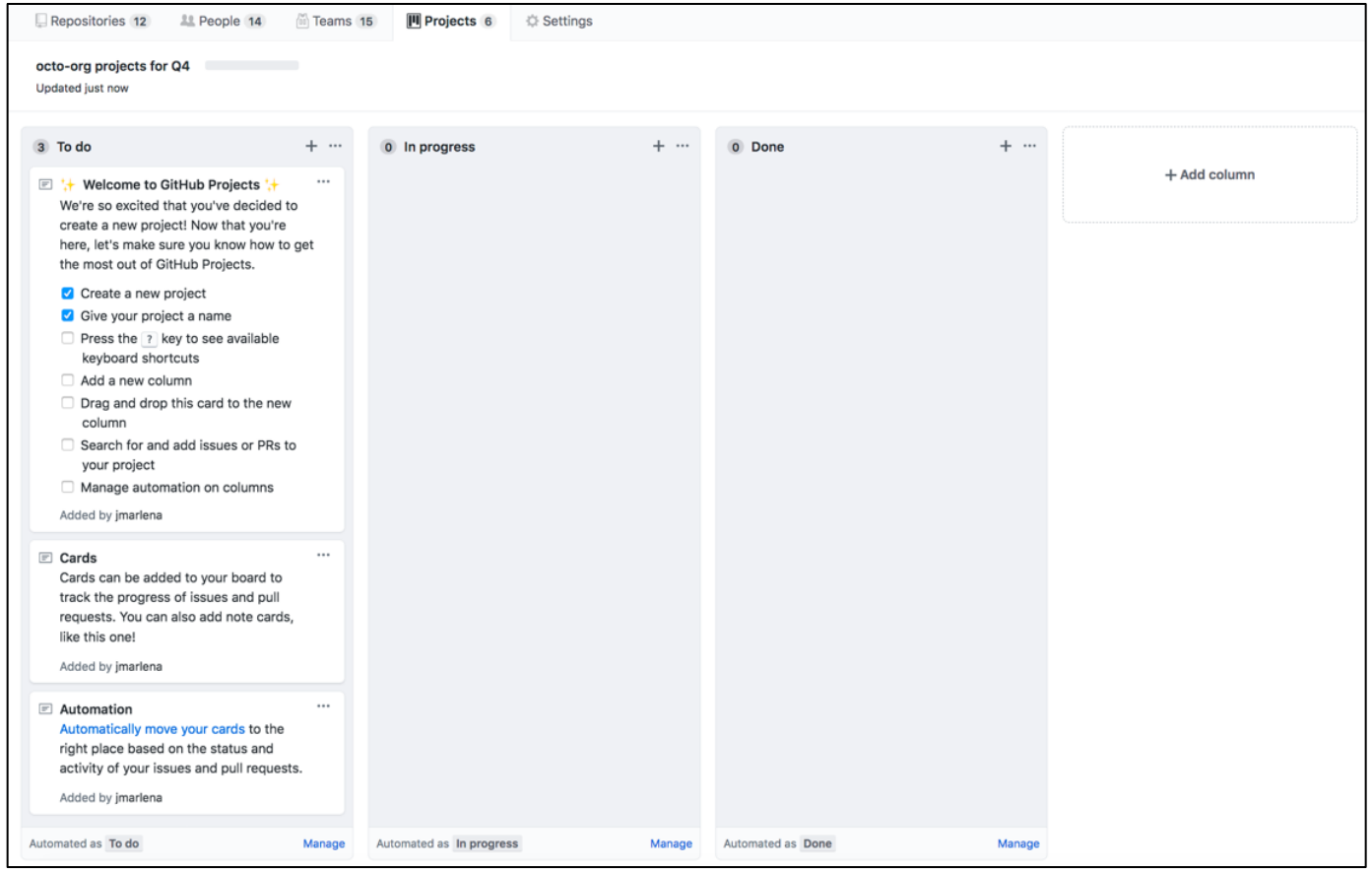
Visualizing work

- Kanban board
  - Product backlog
    - To do/ ice box
    - Comprised of user stories
  - In-progress
  - Done
- Might be unique for Scrum teams

- Image from HBO's Silicon Valley - What's the emergency Erlich!!?? (haha)

Visualizing work

- Kanban board
  - Product backlog
    - To do/ ice box
    - Comprised of user stories
  - In-progress
  - Done
- Might be unique for Scrum teams
- ProjectBoards in GitHub



- Image from HBO's Silicon Valley - What's the emergency Erlich!!?? (haha)

Image refs

- Horse, https://i.kym-cdn.com/photos/images/newsfeed/000/572/093/7dd.jpg
- UML, https://i.pinimg.com/736x/00/b7/15/00b71524a368802ea029ff9a0ff235ea--project-management-geek-humor.jpg
- No code, https://upload.wikimedia.org/wikipedia/en/thumb/f/f3/PearlJam-NoCode.jpg/220px-PearlJam-NoCode.jpg
- Spice data, https://memegenerator.net/instance/69300440/dune-the-data-must-flow
- Zoidberg, http://www.quickmeme.com/meme/3qevhd
- C++, http://weknowmemes.com/tag/you-have-no-class/
- The marvelous Internet, http://knowyourmeme.com/photos/1268028-thanks-to-the-miracle-of-the-world-wide-web
- UML George, https://memegenerator.net/img/instances/500x/55728167/uml-notation-youre-doing-it-wrong.jpg
- Neil, https://thejacketshop.co.uk/neil-diamond-shirt/buy-a-neil-diamond-shirt.jpg
- Being agile, https://www.agilemarketing.net/wp-content/uploads/2017/06/doing-vs-being.png
- Doge, https://i.kym-cdn.com/photos/images/original/000/661/145/819.jpg
- Solid, https://media.tenor.com/images/875e939eae0dc5fed99740541b00ac24/tenor.gif