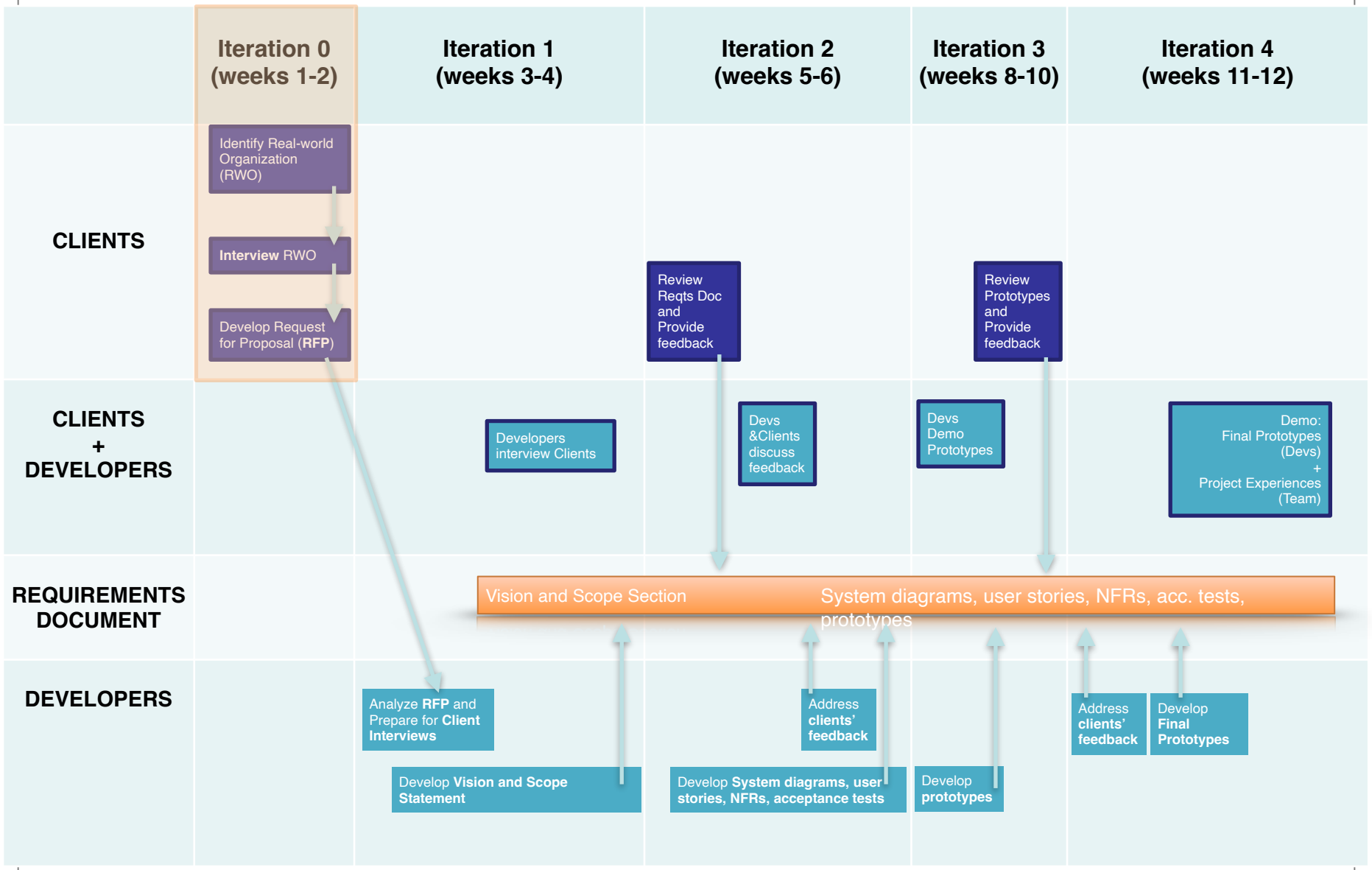
SENG321: Requirements Engineering

Where does this fit?

1. Identify Problem
2. Gather requirements
3. **Analyze & Model requirements**
4. Design solution


Iteration 0 (weeks 1-2)
Iteration 1 (weeks 3-4)
Iteration 2 (weeks 5-6)
Iteration 3 (weeks 8-10)
Iteration 4 (weeks 11-12)
CLIENTS
CLIENTS + DEVELOPERS
REQUIREMENTS DOCUMENT
DEVELOPERS
Identify Real-world Organization (RWO)
Interview RWO
Develop Request for Proposal (RFP)
Review Reqts Doc and Provide feedback
Review Prototypes and Provide feedback
Developers interview Clients
Devs &Clients discuss feedback
Devs Demo Prototypes
Demo: Final Prototypes (Devs) + Project Experiences (Team)
Vision and Scope Section
System diagrams, user stories, NFRs, acc. tests, prototypes
Analyze RFP and Prepare for Client Interviews
Develop Vision and Scope Statement
Address clients' feedback
Develop System diagrams, user stories, NFRs, acceptance tests
Develop prototypes
Address clients' feedback
Develop Final Prototypes

What you will use in your project

In Iteration 2

Use case diagram (include all use cases)

Use Case descriptions (pick 2 main ones), State Charts (if you deem relevant)

Functional requirements (**text**) – could be **User stories** for remaining behavioural descriptions

Non-functional requirements (**text**)

Data models (i.e. Entity relationship diagrams)

All should include Acceptance Tests (Criteria)

In Iteration 3

Domain Model (e.g. Data Flow Diagrams), glossary

UI model

Context: Requirements can be difficult to capture

Not always obvious and have many sources

May not be easy to express clearly in words

Different types at different levels of detail

They are often related to one another

They change and can be time-sensitive

Context: So how do we cope with this?

First we do a good job of gathering/discovering requirements:

- Interviews
- Observation of users
- Analysis of documentation
- Copying from another product's features

Then we **model** and keep track of requirements

Modeling …

The act of representing something with something else

We model reality with text and diagrams

Largely addresses the behavioural aspects of the system -–> functional requirements

Agile vs. “traditional” Modeling

The “agile”: A practical method for modeling to create software systems

Based on best practices

Light-weight

‘just enough’ to get the job done

don’t model for the sake of modeling

Modeling related to Functional requirements

(I) Can be **expressed as TEXT**, e.g.:

Record an immunization event

Manage vaccine inventory, Add a vaccine to inventory

(II) Can be **modeled** in different ways, e.g.:

Behavioral modeling: e.g. Use cases, user stories, state charts

Internal modeling:

Domain and Data modeling: E.g. Entity relationship diagraming

Process modelling: e.g. Data Flow Diagraming

Graphical Interaction modeling: e.g. User-interface modelling

Requirements Engineering: the tension between describing the problem vs. solution

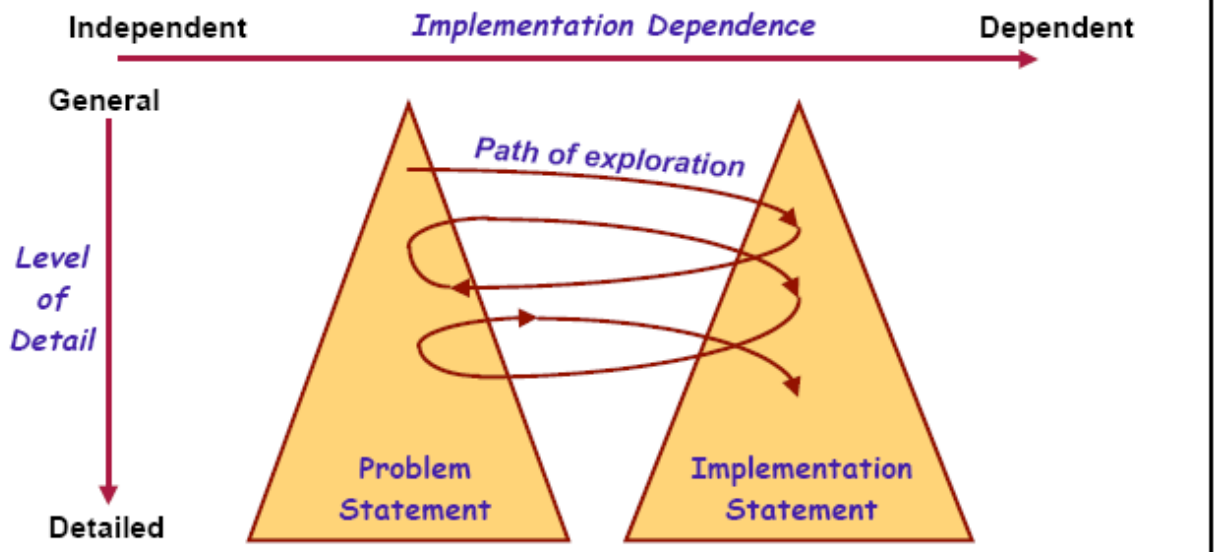


Figure 3: The Twin Peaks model. Exploration of the problem and its possible solutions are closely intertwined. (Adapted from Moffett 1999)

➔ Several approaches, waterfall vs. agile development at two extremes

Let's see these through some examples

(I) Functional requirement(s): in **TEXT**

Record an immunization event

Manage vaccine inventory, Add a vaccine to inventory

(II) Can be **modeled** in different ways, e.g.:

Behavioral modeling: e.g. **Use cases**, user stories, state charts

Internal modeling:

Domain and Data modeling: E.g. Entity relationship diagraming

Process modelling: e.g. Data Flow Diagraming

Graphical Interaction modeling: e.g. User-interface modelling

Use Case Modeling

Diagram(s) provide overview of actors and use cases

The work is in the textual use case specifications

- Can be done at different levels of detail
- Can be done with varying formality

Use Case Modeling

Informal, e.g.:

The practitioner selects a patient to immunize and reviews the patient' vaccine history. They look for possible allergies & previous adverse events. They confirm that the patient is eligible for the vaccine. Next they administer the vaccine, recording the dose and lot number as well as date, time, location, and any adverse events.

Use Case Modeling

More formal

- Preconditions
- Success steps
- Post condition(s)
- Alternate paths

Use Case Modeling

Preconditions: The patient has been selected as part of an audit group or located as an individual

Success steps:

Scan patient's history
Verify no allergies or adverse events
Verify eligibility (ref. Eligibility business rules)
Administer vaccine:
    Record …

Success post condition: vaccine delivered and recorded

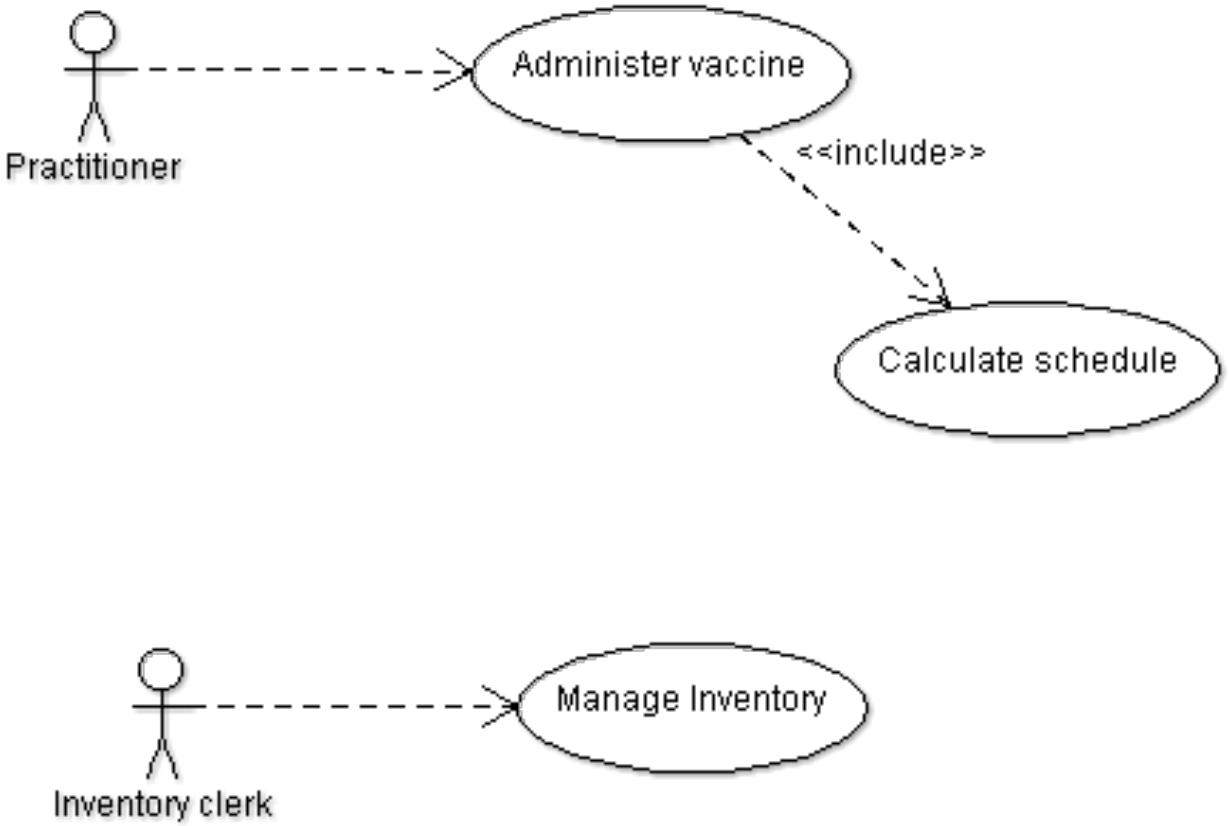
Alternate paths:

a) Patient contraindicated – record reason
b) Patient ineligible – record reason

Functional requirement(s): in **TEXT**

Record an immunization event
Manage vaccine inventory, Add a vaccine to inventory

Use Case Modeling: Use Case Diagram



Let's see these through some examples

(I) Can be **expressed as TEXT**, e.g.:

Record an immunization event

Manage vaccine inventory, Add a vaccine to inventory

(II) Can be **modeled** in different ways, e.g.:

Behavioral modeling: e.g. Use cases, **user stories**, state charts

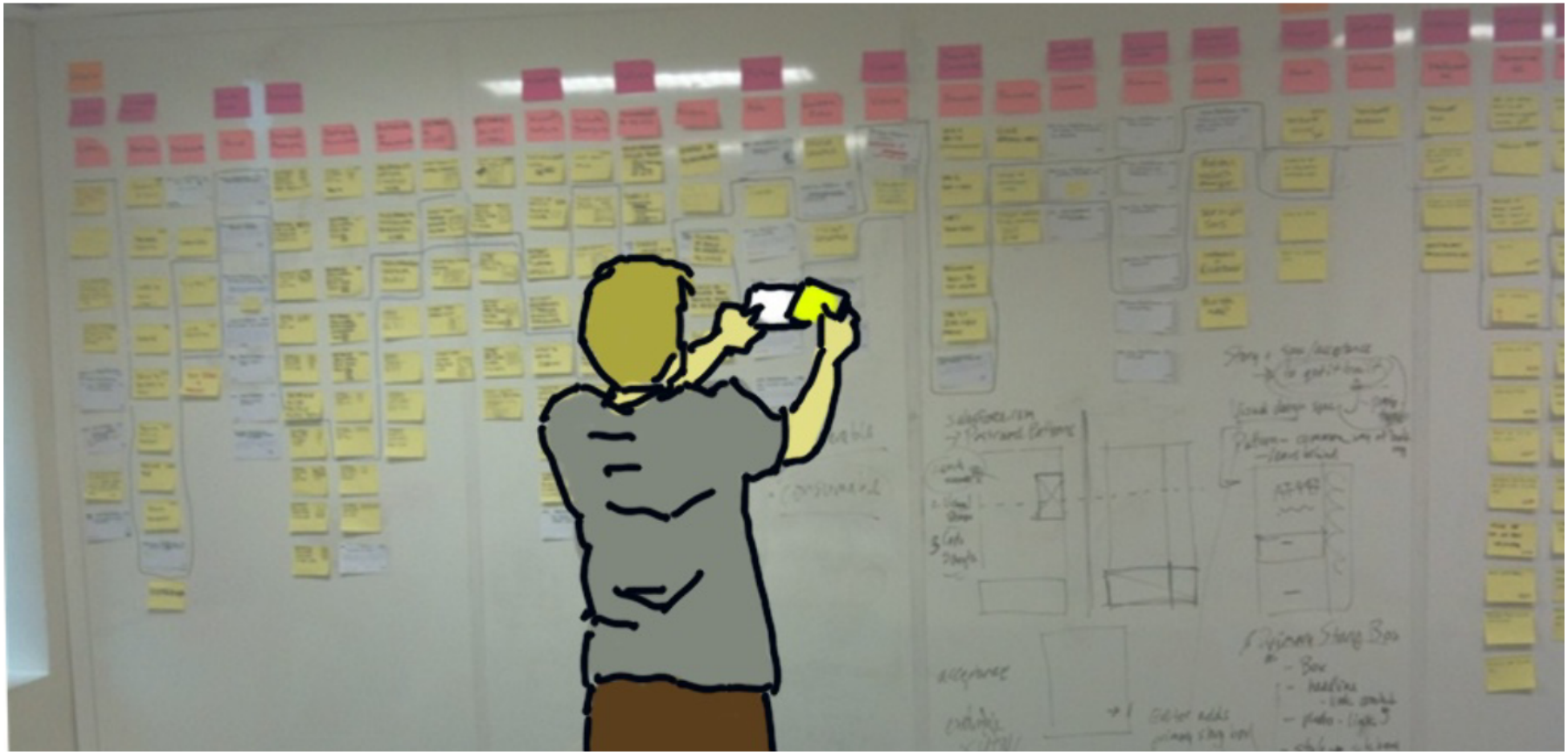
Internal modeling:

Domain and Data modeling: E.g. Entity relationship diagraming

Process modelling: e.g. Data Flow Diagraming

Graphical Interaction modeling: e.g. User-interface modelling

User Stories



User Stories

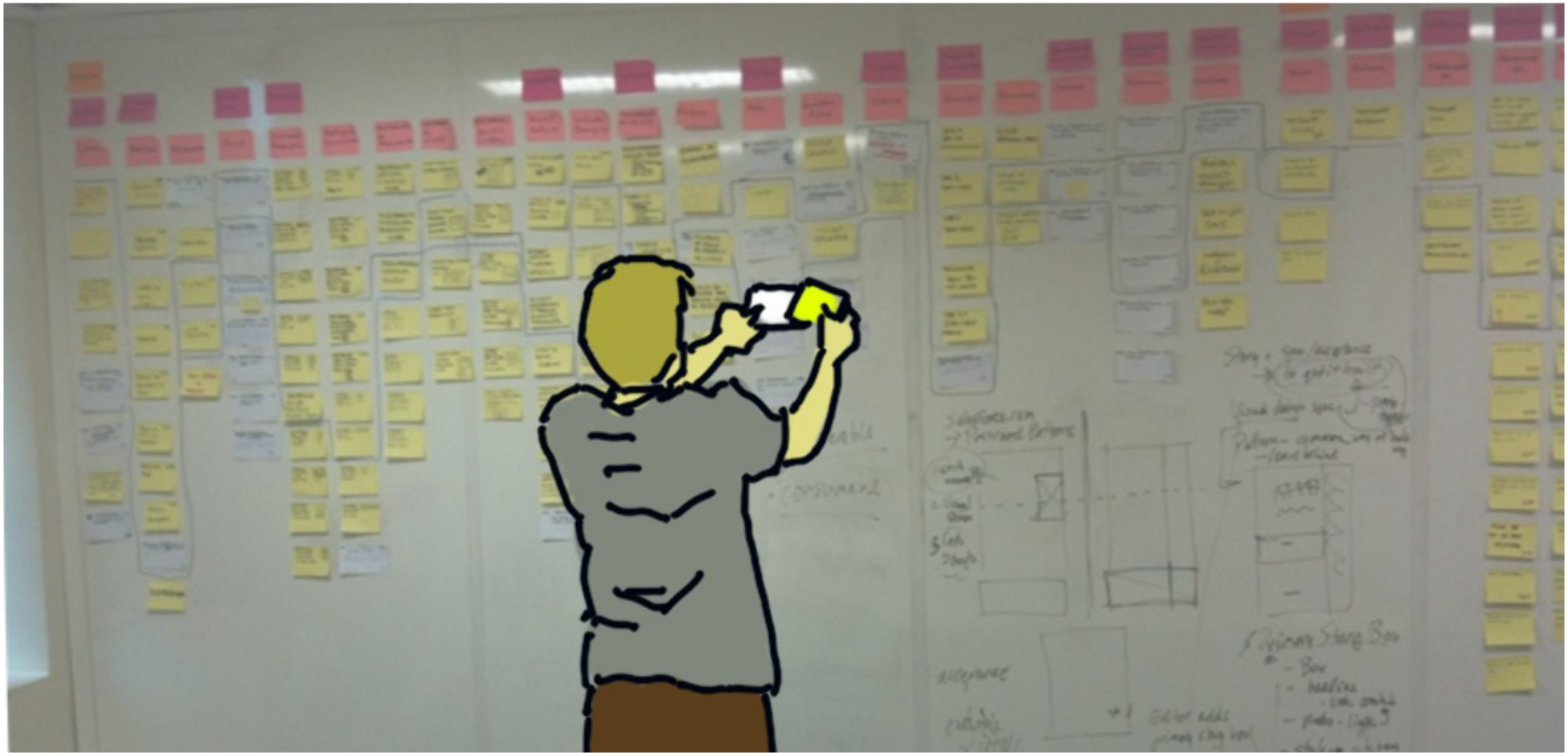
As a <**type of user**>, I want <**some goal**> so that <**some reason**>

examples:

As a ***small business owner***, I want to ***create an invoice*** so that ***I can bill a customer***.

As a ***customer***, I want to ***update my customer profile*** so that ***future purchases are billed to a new credit card number.***

User Stories– easy to organize in a Development Roadmap



Acceptance Criteria for User stories

Define the minimum conditions

(typically) The Product Owner accepts user story completion when all acceptance criteria for a user story is met

Examples

As a ***small business owner,*** I want to ***create an invoice*** so that ***I can bill a customer.***

acceptance criteria:

A small business owner can access the invoice form

A small business owner can create a new invoice

A small business owner receives a notification after submitting the new invoice

Examples

As a ***customer,*** I want to ***update my customer profile*** so that ***future purchases are billed to a new credit card number.***

A customer can access the profile page

A customer can input credit card information

A customer is charged on updated credit card information for new purchase

Let's see these through some examples

(I) Functional requirement(s): in **TEXT**

Record an immunization event

Manage vaccine inventory, Add a vaccine to inventory

(II) Can be **modeled** in different ways, e.g.:

Behavioral modeling: e.g. Use cases, user stories, **state charts**

Internal modeling:

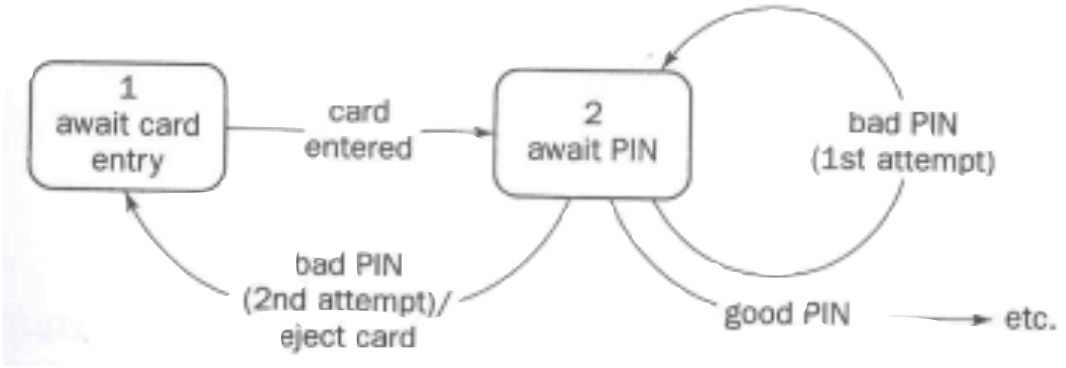
Domain and Data modeling: E.g. Entity relationship diagraming

Process modelling: e.g. Data Flow Diagraming

Graphical Interaction modeling: e.g. User-interface modelling

State Charts (or State diagrams)

Model the system's behaviour as a sequence of states in response to triggers/ actions



Let's see these through some examples

(I) Functional requirement(s): in **TEXT**

Record an immunization event

Manage vaccine inventory, Add a vaccine to inventory

(II) Can be **modeled** in different ways, e.g.:

Behavioral modeling: e.g. Use cases, state charts

Internal modeling:

Domain and Data modeling: E.g. **Entity relationship diagraming**

Process modelling: e.g. Data Flow Diagraming

Graphical Interaction modeling: e.g. User-interface modelling

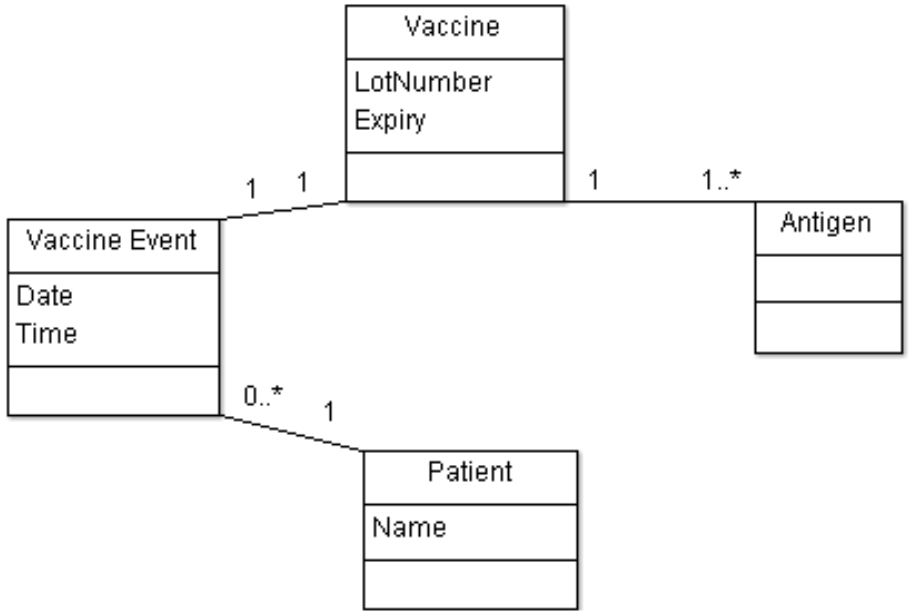
Internal modeling: Domain and Data

Domain models Support the functional requirements

Models things and relationships in the **problem domain**

Includes **glossary**/data dictionary for details

Data Model: Entity Relationship Diagram



Vaccine: A specific instance of vaccine, containing 1 or more antigens and having an assigned lot number and expiry date.

Let's see these through some examples

(I) Functional requirement(s): in **TEXT**

Record an immunization event

Manage vaccine inventory, Add a vaccine to inventory

(II) Can be **modeled** in different ways, e.g.:

Behavioral modeling: e.g. Use cases, state charts

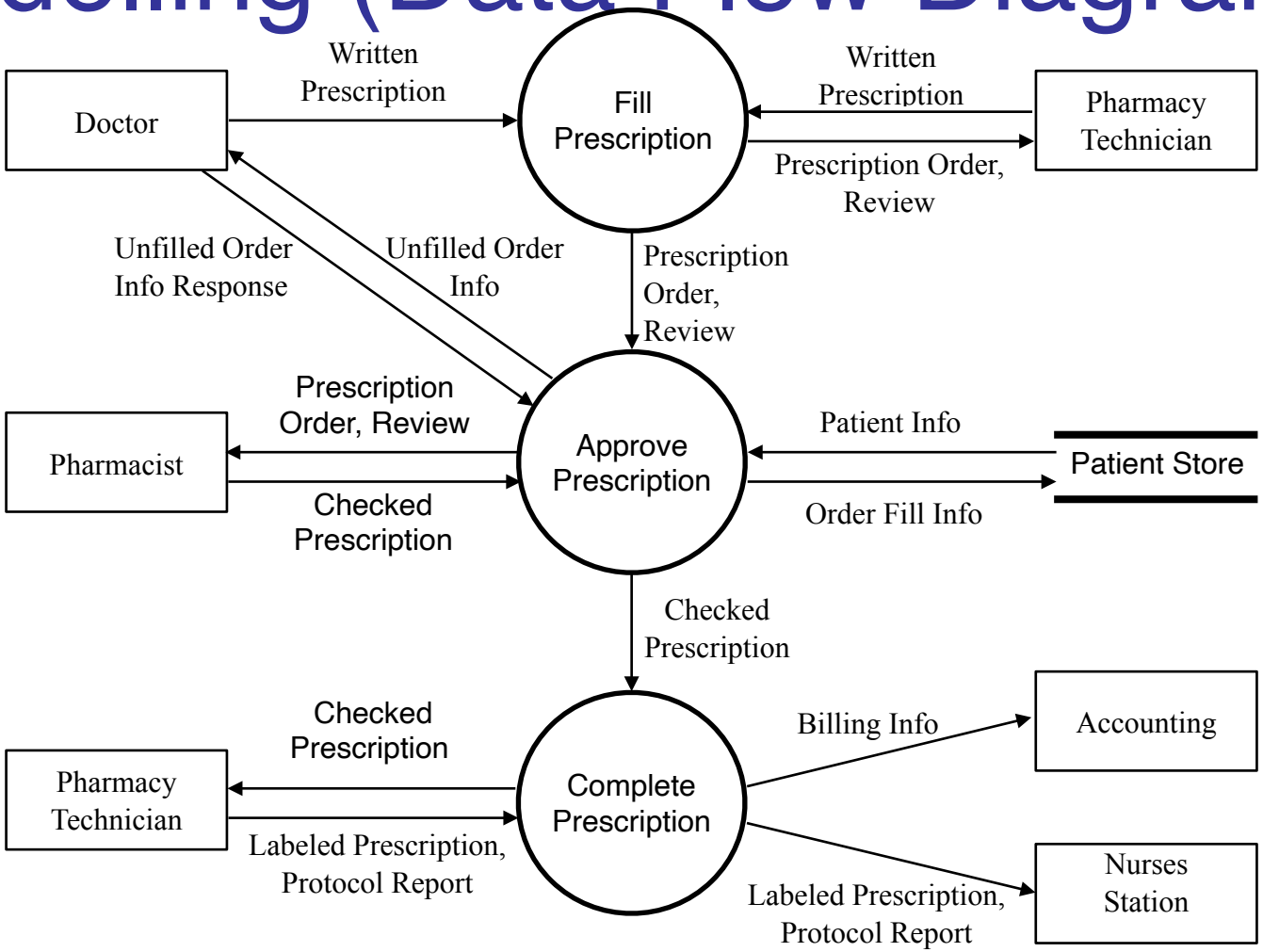
Internal modeling:

Domain and Data modeling: E.g. Entity relationship diagraming

Process modelling: e.g. **Data Flow Diagraming**

Graphical Interaction modeling: e.g. User-interface modelling

Internal modelling: process modelling (Data Flow Diagrams)



Internal modeling: process modelling (Data Flow Diagrams)

Brainstorm a Data Flow diagram for our vaccine administration example?

Let's see these through some examples

(I) Functional requirement(s): in **TEXT**

Record an immunization event

Manage vaccine inventory, Add a vaccine to inventory

(II) Can be **modeled** in different ways, e.g.:

Behavioral modeling: e.g. Use cases, state charts

Internal modeling:

Domain and Data modeling: E.g. Entity relationship diagraming

Process modelling: e.g. Data Flow Diagraming

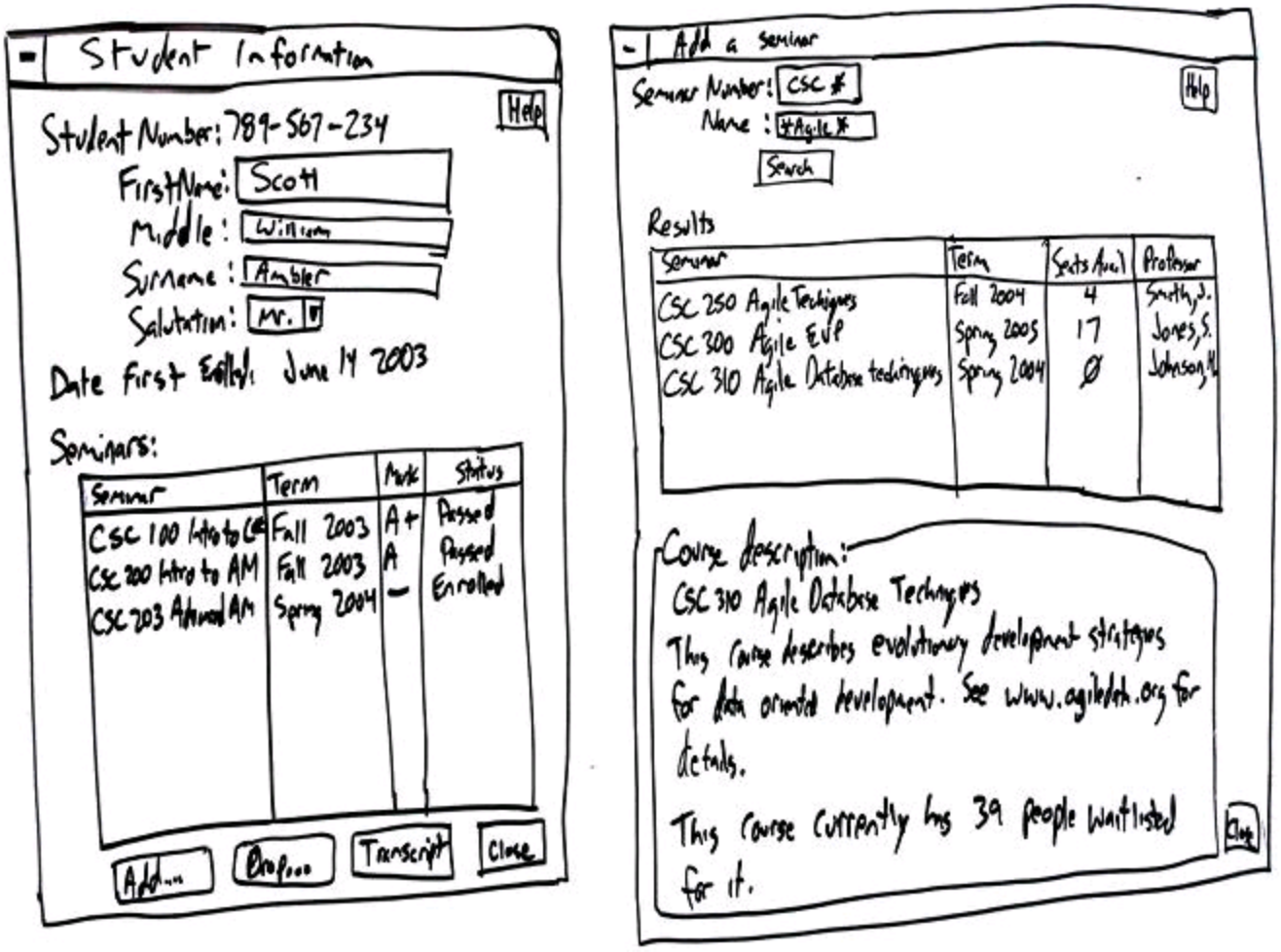
Graphical Interaction modeling: e.g. **User-interface modelling**

Graphical User Interface Modeling

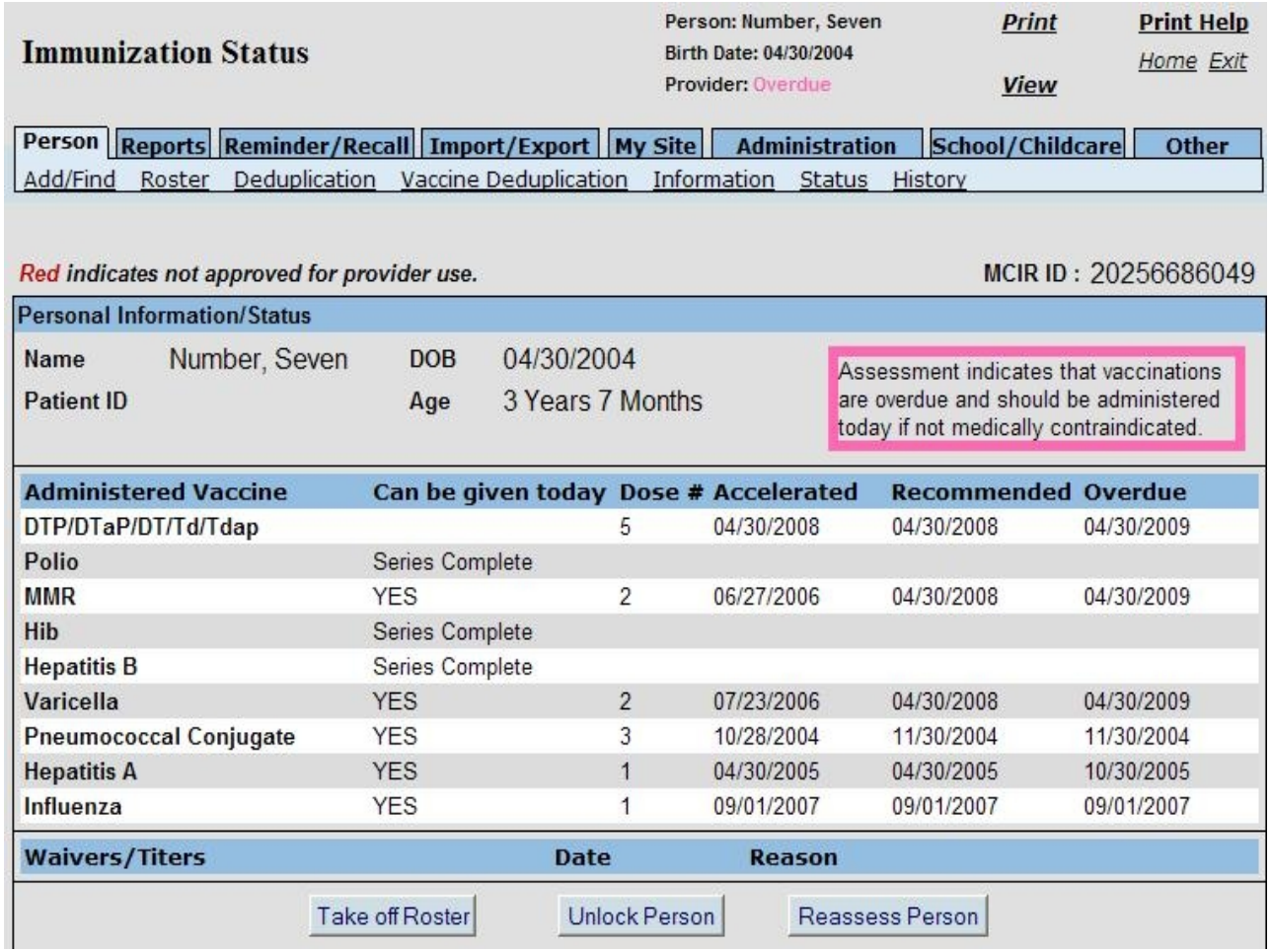
Paper or electronic prototypes

Model essential parts of the user interface

E.g. UI paper prototype



UI HTML prototype



Immunization Status

Person: Number, Seven
Birth Date: 04/30/2004
Provider: Overdue

Print *View*

Print Help *Home* *Exit*

Person | **Reports** | **Reminder/Recall** | **Import/Export** | **My Site** | **Administration** | **School/Childcare** | **Other**

Add/Find Roster Deduplication Vaccine Deduplication Information Status History

Red indicates not approved for provider use.

MCIR ID : 20256686049

Personal Information/Status

Name Number, Seven **DOB** 04/30/2004
Patient ID **Age** 3 Years 7 Months

Assessment indicates that vaccinations are overdue and should be administered today if not medically contraindicated.

Administered Vaccine	Can be given today	Dose #	Accelerated	Recommended	Overdue
DTP/DTaP/DT/Td/Tdap		5	04/30/2008	04/30/2008	04/30/2009
Polio	Series Complete				
MMR	YES	2	06/27/2006	04/30/2008	04/30/2009
Hib	Series Complete				
Hepatitis B	Series Complete				
Varicella	YES	2	07/23/2006	04/30/2008	04/30/2009
Pneumococcal Conjugate	YES	3	10/28/2004	11/30/2004	11/30/2004
Hepatitis A	YES	1	04/30/2005	04/30/2005	10/30/2005
Influenza	YES	1	09/01/2007	09/01/2007	09/01/2007

Waivers/Titers	Date	Reason

Take off Roster | Unlock Person | Reassess Person

What you will use in your project

In Iteration 2

Use case diagram (include all use cases)

Use Case descriptions (pick 2 main ones), State Charts (if you deem relevant)

Functional requirements (**text**) – could be **User stories** for remaining behavioural descriptions

Non-functional requirements (**text**)

Data models (i.e. Entity relationship diagrams)

All should include Acceptance Tests (Criteria)

In Iteration 3

Domain Model (e.g. Data Flow Diagrams), glossary

UI model