COMP2003 Project Execution

Dr Ji-Jian Chin

Module Leader

UNIVERSITY OF PLYMOUTH

Session outline

- Requirements Analysis
- Design and Development
- Security
- Testing

UNIVERSITY OF PLYMOUTH

Initial Client Meetings

- Briefs out by 16 October. Meet & Greet at 11am on 20 October.
- After that, arrange meeting with client at their convenience.
  - Initial requirements gathering, to be translated into product backlog.
- All expected to go
  - If you missed out or misunderstood an important point, others should be listening actively as well. 3 or 4 pairs of ears better than 1!
- If you record the meeting, check that’s ok with the client! Recordings may be valuable to revisit important points.
- Document the meeting with minutes!

UNIVERSITY OF PLYMOUTH

Categories of requirements

Quick reminder from last week

- Functional vs non-functional
- Safety and security (system security)
- Quality, domain, interoperability, CRUD
- High level vs detailed

Functional

- Functional requirements
  - The statements of services that the system should provide
  - How the system should react to particular inputs

UNIVERSITY OF PLYMOUTH

Refining

- From the high level, need to consider more detail
- Maybe not quite enough to write software from
- Need to add in what it will NOT do

Requirements Refinement
System must be able to respond to an invalid user-id / password. The system should display the error message X. It should not allow access to the system

Non-functional

- Constraints on system services
  - Reliability, availability, performance, efficiency, usability
- External requirements
  - Social, legal, ethical, interoperable

Non-functional examples

- Performance criteria such as desired response times for updating data in the system or retrieving data from the system
- Ability of system to cope with high level of simultaneous access by many users
- Availability of system with the minimum of downtime
- Time taken to recover from a system failure
- Anticipated volumes of data, either in terms of transaction throughput or of what must be stored
- Security considerations such as resistance to attacks and the ability to detect attacks

Usability Requirements

- Those requirements that will ensure there is a good match between the system developed and the users being able to achieve their goals when using the system
- International Organisation for Standardisation (ISO) definition:
  - “The extent to which specified users can achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use”

Usability examples

- After 30 minutes of training a new user will be able to perform.. In less than 15 minutes with less than 5% error rate
- The system will present X data in centimetres
- Users will give an 85% average satisfaction rating on the user satisfaction questionnaire
  - Many aspects of user interface are subjective or intangible. You could use a questionnaire to provide a quantitative assessment
- Usability is testable! Not vague notion of user-friendliness

Functional vs usability

- Functional
  - The library counter staff should be able to process book loans
- Usability
  - An experienced member of the library counter staff should be able to process a normal book loan with no error and within 30 seconds (95%) of the time

Legal

- A number of possible legal requirements
- GDPR
- Disability discrimination act
- Computer Misuse Act
- Sarbanes-Oxley
- Obscene publications
- Sale of …. To minors

A mistake often made by novices is that they think there is nothing that applies to their project/software

Social & Ethical

- Requirements may relate to:
  - Privacy
  - Safety
  - Decency
  - Pollution
  - Sustainability

UNIVERSITY OF PLYMOUTH

Interoperability requirements

- System may need to provide a service to a pre-existing system
- The system may need to employ a service provided by a pre-existing system
- The system may need to employ existing data stores and therefore be constrained by the format of these.

Non-Functional Requirements

- Technical : Web based application interacting with lights in SMB109. Data hosted on remote server provided. Interface for student is browser based but for Lecturer is mobile.
- Performance : Response times not within scope.
- Usability : Interfaces to conform to accessibility rules
- Reliability : Outside of scope
- Security : OWASP top 10 to be addressed where appropriate. Lecturers log in but students do not. Name info only for lecturer

UNIVERSITY OF PLYMOUTH

Finding it all out

- Background reading
- Interviewing
- Observations
- Document sampling
- Questionnaires
- Focus groups

UNIVERSITY OF PLYMOUTH

Questions to ask

- To understand the operations and processes
  - Ask ”What do you do?”
- Understand how operations should be performed
  - Ask “How do you do it?”. “What steps do you follow?”
- Understand what information is needed to perform these operations
  - Ask ”What information do you use?”, “What forms or reports do you use?”

UNIVERSITY OF PLYMOUTH

Confusing user requirements

- To assist customers in identifying appropriate items for purchase whilst browsing the online store, the system will provide a facility whereby the customer can view recent purchases (either for the last month, last quarter, or last year). Initially this facility is off, but may be turned on/off at any point during the browsing session, and can be toggled between last month, last quarter and last year at any time…

A functional requirement

UI issues

Too much detail!

Re-written

Note the numbered requirement

- 2.4.5 Recent purchase view
  - The system shall provide a facility whereby customers can view recent purchases whilst browsing the online store
  - *Rationale*:
    - This facility helps the customer to make better purchasing choices
  - *System requirements reference*:
    - OnlineStore/Section 4.1
  - *Proposer*:
    - Fred Bloggs, Plymouth Office

Writing Techniques

- Simple structured text
- Use diagrams
  - Use case descriptions describe actor-system interactions
  - Sequence and communication diagrams to describe object interactions
  - Statecharts to describe state-dependent behavior
  - Techniques for specifying algorithmic requirements

UNIVERSITY OF PLYMOUTH

Requirements elicitation

- Is working with stakeholders to find out about the application domain and requirements
  - Activities in the application domain may yield system services
  - Things in the domain may need to be represented in the system

UNIVERSITY OF PLYMOUTH

Elicitation : Viewpoints

- Stakeholders have different viewpoints on the system and will see the problem in different ways
  - They will structure the elicitation process and requirements themselves
- A win condition for a particular stakeholder is a requirement that is considered critical by that stakeholder
- Conflicting win conditions are potential problems/risks

Elicitation : Interviews

- Different stakeholders can provide differing types of information
  - High level management : the strategic view
  - Middle management : Overview of business process
  - Operational management and users : details of the business processes, operational problems
- Users sometimes get stuck in the details of how their job is currently done
- Plan carefully how you need to speak to, why and when
  - They will not thank you for wasting their time
  - Do not spend time asking for things in interviews that can be gained from elsewhere

UNIVERSITY OF PLYMOUTH

Conducting interviews

- Prepare yourself and your interviewee beforehand
  - What information are you expecting from them?
- Record the contents of the interview
- Summarise the points made in order to ensure that they have been understood – clarify facts vs opinions
- Provide at least some opportunity for the interviewee to raise issues of their own
- Ask ”who else can provide information”
- Do NOT promise to include something they ask for in the system
- Afterwards write it up, get the interviewee to confirm the details and clarify if the notes can be made public

UNIVERSITY OF PLYMOUTH

Elicitation : Questionnaires

- Often used when number of stakeholders is too large
  - Invite all or a sample to complete the questionnaire
  - Only a fraction will actually respond
- Web-based questionnaires are easier to construct and analyse
- Closed questions are most commonly used and the data most easily analysed
- Questions must be crystal-clear as there is limited opportunity for clarification
- Consider very carefully in advance what information is required

UNIVERSITY OF PLYMOUTH

Elicitation : Observation

- Observation of the use of existing systems can provide information about possible system services and wider business processes
- Observation of the use of prototypes can help to elicit or validate requirements for the intended system
- Can be used to check validity of information from other sources eg interviews.
- Captures information that is not consciously held by stakeholders

Elicitation : Document Analysis

- Document analysis can provide information about the existing system and domain
  - Policy manuals, IT systems documentation, business process documents.
- BUT documentation often gets out of date

Elicitation : Event modelling

- What events does the system need to respond to?
- Events are either flow, control or temporal (time-based) depending on how the system becomes aware of the event
  - Flow oriented – incoming data flow
  - Control - incoming control flow
  - Temporal – specified time
- Customer order (flow)
- Distribution company provides music information update (flow)
- Status report requested by store manager (control)
- Produce end-of-month sales summary report (temporal)

Elicitation : Business analysis

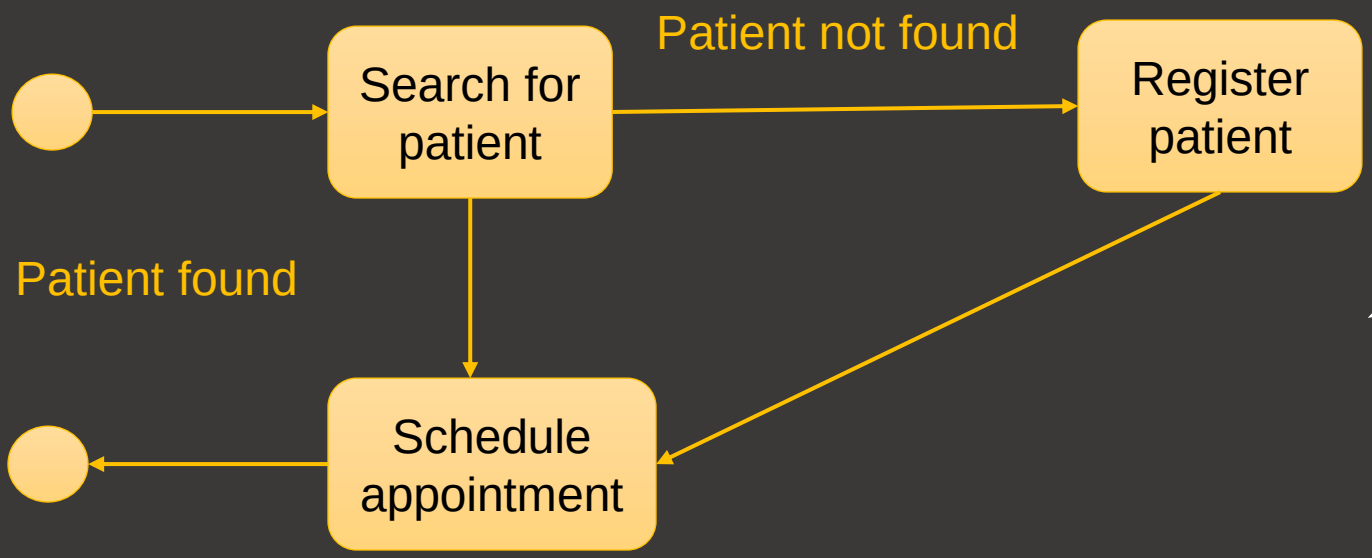
- Consider the business need – their problems and root causes
- “Currently the customer service division does a lot of manual work. Each person in the team spends more than a day per week dealing with paperwork, when their primary focus should be customer interaction. Tracing orders, queries and complaints is difficult, and a lot of time is spent manually pulling information together for reporting purposes. When answering a customer call, the customer service agent is not able to see al recent orders, queries and complaints of the current customer, so customers have to repeat information depending on who they get to talk to .. “

UNIVERSITY OF PLYMOUTH

Business needs are not system requirements. Business objectives, project objectives and/or system requirements may be derived from the business needs

Elicitation : Business process modelling

- Dentist's appointment system. Event = Patient requests appointment



Yields 4 requirements

1. Book patient appointment
2. Search for Patient
3. Register Patient
4. Schedule appointment

UNIVERSITY OF PLYMOUTH

Requirements management

- Business managers and users will happily:
  - Ask for every possible requirement they can think of
  - Change their requirements as and when they see fit
- They do not consider cost, time, effort or your sanity..
- Unless managed this can sink your project
  - Ensure requirements can be traced back to business objectives
  - Employ prioritization, staged delivery, cost estimation and change control

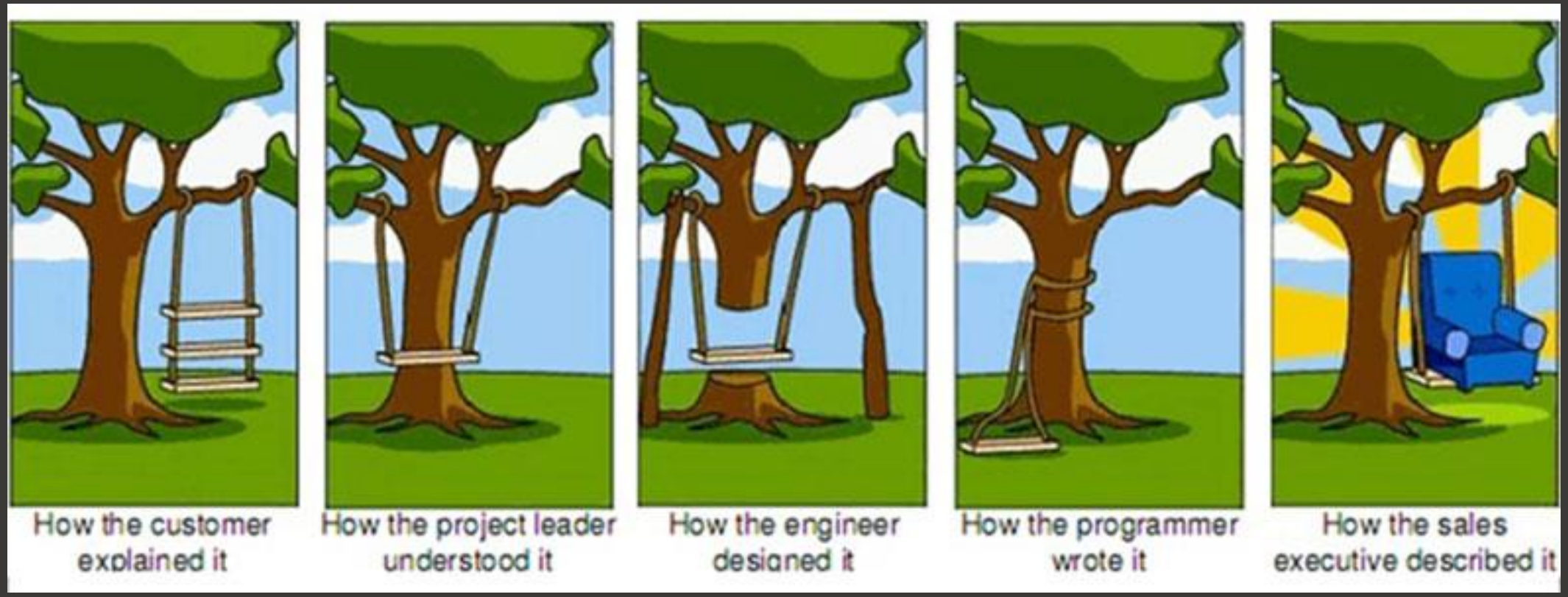
UNIVERSITY OF PLYMOUTH

Expressing requirements priorities

- Use a simple numeric scale
- MoSCoW is sometimes seen in literature
  - Must be included in current delivery
  - Should be included, but not necessarily in current delivery
  - Could – would be nice
  - Won't – at least in current delivery
- Delivery could be a 2 week window (sprint)

Recording information

- So you have gathered those requirements
  - How did you record them?
- Need a mixture of diagrams, data and text
- You need a model
- Models describe systems efficiently and elegantly
- A picture speaks a thousand words!



UNIVERSITY OF PLYMOUTH

Requirements

- Think about what you know about the activities or the tasks being carried out
  - This is part of requirements modelling
- Think about what you know about the essential elements of the requirements
  - This goes into the requirements model

UNIVERSITY OF PLYMOUTH

Requirements -> Analysis

- Analysis activities use the documents output from the requirements gathering
- Identify what the system must do
- An analyst seeks to understand the organisation, requirements and objectives
- Designer will seek to specify a system that fits with organisation, provides requirements effectively and assists in meeting objectives

UNIVERSITY OF PLYMOUTH

Purpose of analysis

- Analysis aims to identify
  - A software structure that can meet the requirements
  - Common elements among the requirements that need only be defined once
  - Pre-existing elements that can be reused
  - The interaction between different requirements

Analysis & Design

- Design
- Develop an architectural structure for software components, DB, user interface
- Low level design – develop detailed tables, classes, methods and structures
- Storyboards are part of this
  - Design for each aspect of user interface
  - Each dialog or goal

The Analysis model

- The model must confirm what users want a new system to do
- It must be:
  - Understandable for users
  - Correct scope
  - Correct detail
  - Complete
  - Consistent between different diagrams and models

What an analysis model does

- Describes what the software should do
- Represents people, things and concepts important to understand the system
- Shows connections and interactions among these people, things and concepts
- Show the business situation in enough detail to evaluate possible designs
- Is organized/structured so it can be useful for designing the software

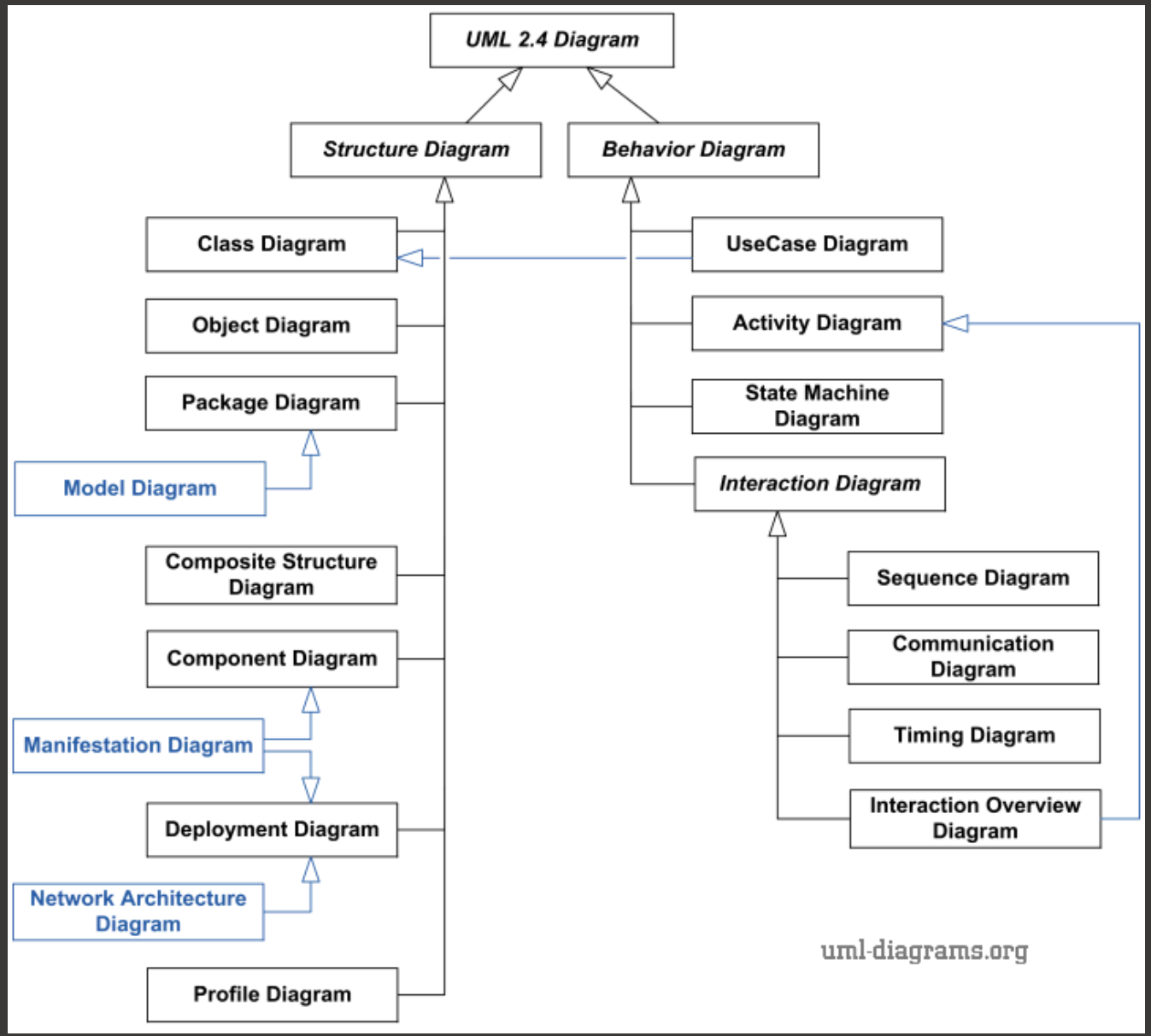
UNIVERSITY OF PLYMOUTH

Modelling

- Mapping : a model is always an image, a representation
- Reduction : a model does not capture ALL attributes – only those that are relevant
- Pragmatism : It needs to be useful. To be useful consider who it is for, why it is being created and what will it be used for.

UML

- Unified Modelling Language
- A comprehensive set of diagrams that together represent a model of the system



UML Models/Diagrams

- Requirements are modelled with a Use Case Diagram, supporting user stories and descriptions

Use Case is an activity that a system performs in response to a request from a user (AKA functional requirement)

UNIVERSITY OF PLYMOUTH

User story

- One short sentence in the every day language of the end user that states what a user does as part of their work
- Maps directly to one use case

Template

As a <role played> I want to <goal desire> so that <reason or benefit>

As a Checkout Desk Librarian I want to issue loan items so that I can serve more members quickly

UNIVERSITY OF PLYMOUTH

Use Case Diagram : purpose

- Document the functionality of the system from the users' perspective
- Document the scope of the system
- Document the interaction between the users and the system using supporting use case descriptions

Requirements Validation

- User story must have acceptance criteria
- 1. Member look-up must be by name or member number
- 2. Current loan items and member status must be displayed

Identifying User stories and Use Cases

- Work out who you need to speak to
- Identify all potential users for the new system
- Classify the potential users in terms of the functional role (stock control, manage members)
- Further classify potential users by organisation level (operational, management, executive)

UNIVERSITY OF PLYMOUTH

User stories .. continued

- Should now have a list of preliminary use cases organized by type of user
- Look for duplicates with similar names, resolve inconsistencies
- Identify where different types of users need the same thing
- Review list with stakeholders

User	User goal and resulting use case
Potential customer	Search for item Fill shopping cart View product rating and comments
Marketing manager	Add/update product information Add/update promotion Produce sales history report
Shipping personnel	Ship order Track shipment Create item return

UNIVERSITY OF PLYMOUTH

Event decomposition technique

- Consider the types of event in the system environment that requires a response from the system
- For each external event identify and name the use case
- Consider temporal events that require a response from the system
- For each temporal event, identify and name the use case, establish a point of time that triggers the use case
- Consider the state event that system might respond to

User stories

- SHOULD NOT focus on how to deliver the request but on what happens
- Customer value is focus of the story
  - It is in their language, not the technical
- Stories must be able to be estimated, or they are too big
  - Break it down
- User story ought to be delivered within the 2 week sprint – so it should be small
- User story should be testable - Review theories on Unit testing – links on DLE

UNIVERSITY OF PLYMOUTH

UNIVERSITY OF PLYMOUTH

Development Phase

Sprints, Trello and Git

UNIVERSITY OF PLYMOUTH

Sprint Details

- After initial meeting with client, you should be holding regular sprint stand ups every other day, or at least once a week.
- What do you do during those periods?

UNIVERSITY OF PLYMOUTH

Choose your tech stack

- Do you understand your project?
- What is your time to an MVP/TTM?
- How fast do you need to grow? – horizontal/vertical
- How will you maintain your stack?
- What is your budget/resources? – this will be your time

UNIVERSITY OF PLYMOUTH

Sprint Details

- Review client requirements
- Determine possible solutions and assign members to explore these solutions
- Learn new technologies fast! (LinkedInLearning, Youtube, other Git repos, official learning content from open source libraries). For code, fastest way is still to experiment.
- Get suggestions and templates from AI but do additional research for evidence to back or disprove those suggestions.

UNIVERSITY OF PLYMOUTH

Sprint Details

- Create a board on Trello to facilitate Scrum.
- Start with product backlog
- Then create sprint backlog
- Assign tasks to members, discuss results during review
- Take snapshot end of every sprint and place on repo.
- Additional features, check out Trello - YouTube

Sprint Details

- Development platforms – choose your IDE according to the project requirements (Android Studio, Xamarin, Powerapps, VS, Unity/Unreal)
- Get familiar in the first 2-3 weeks. Write ‘hello world’ programs to get started, then research modules you need to use.
- Youtube, ChatGPT, Stackoverflow, LinkedIn Learning are good resources!
- Allocate time to learn this new tech to fulfil solutions.
- Allocate members to explore different tech stacks which may fit the solution. Gather your results and findings at end of Semester 1’s Sprint 2.

UNIVERSITY OF PLYMOUTH

Sprint Details

- Code repository: Github Classroom
- Upload your evidences here

UNIVERSITY OF PLYMOUTH

Security

UNIVERSITY OF
PLYMOUTH

Security Considerations

- IAAA – identification, authentication, authorization and auditing
- Data protection – data at rest, data in transit
- Input validation – security against XSS and CSRF
- Secure coding practices – avoiding common vulnerabilities (OWASP TOP 10)
- Session management – how do you handle secure tokens and cookies?
- Error handling – do not reveal sensitive information in error messages/logs

UNIVERSITY OF PLYMOUTH

Security Considerations

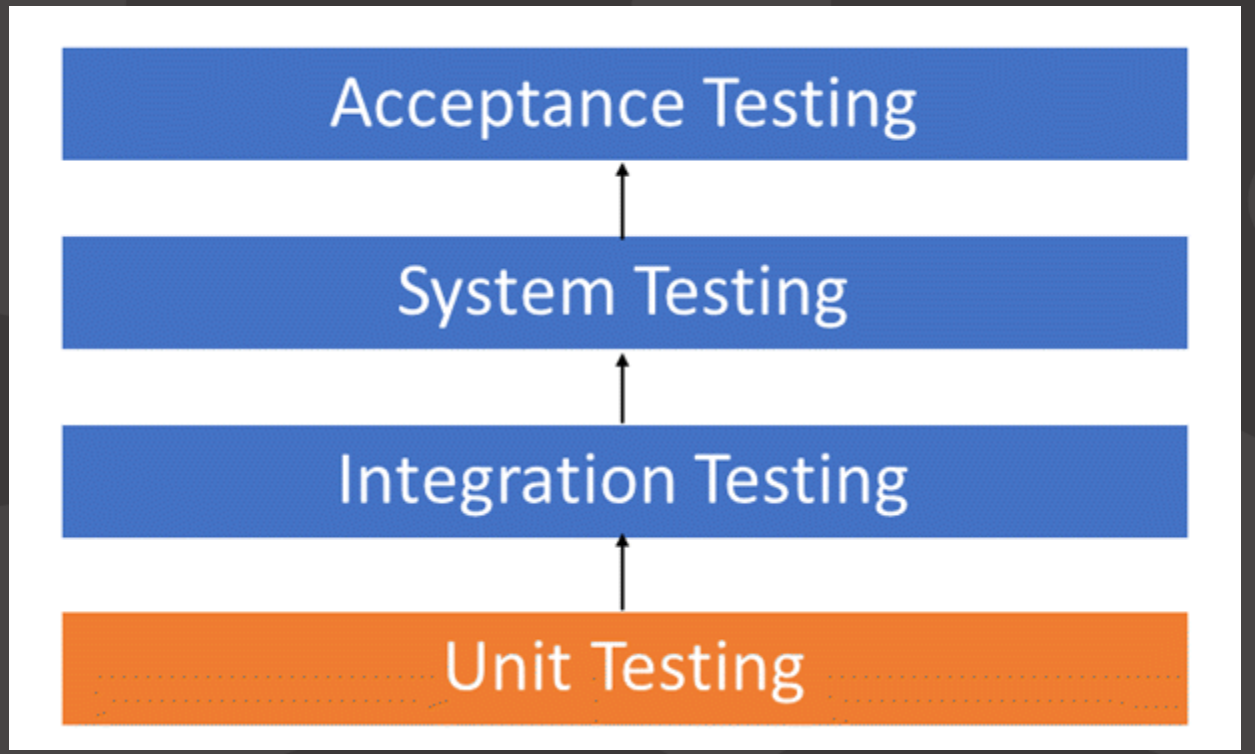
- IAM – identity and access management, DAC or RBAC?
- CORS – cross origin resource sharing
- Security Headers
- Security updates – how will you handle CVEs? Patches?
- Training – user education on how to properly use your app
- API security - APIsec University
- Secure code review

UNIVERSITY OF PLYMOUTH

Testing

UNIVERSITY OF
PLYMOUTH

Types of SE Tests



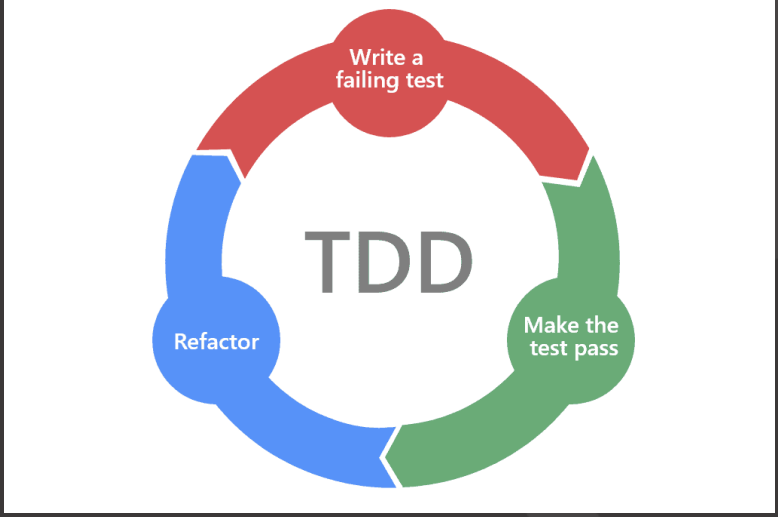
Unit Testing

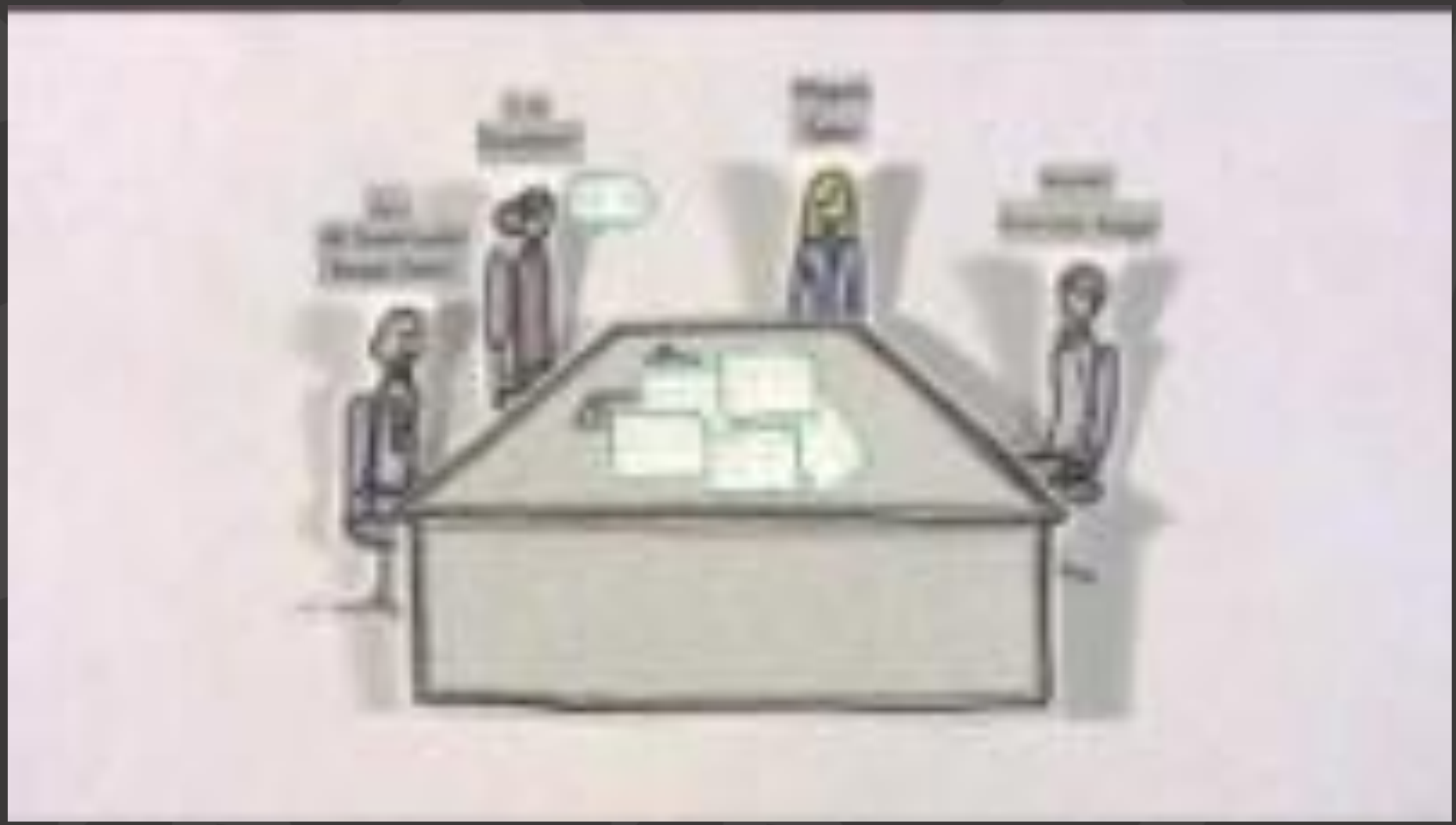
- Done per user story
- Comes in the later half of the sprint once product is developed
- Write test scripts for execution.

UNIVERSITY OF PLYMOUTH

Test Driven Development (TDD)

- Extensive use of testing frameworks.





TEST-DRIVEN DEVELOPMENT

in 3 minutes

Integration Testing

- Test the whole software module instead of individual units
- focuses on evaluating the interactions and connections between different components or modules of a software application
- aims to ensure that these components work together as intended and can seamlessly exchange data and information.

Types of Integration Testing

- Big bang – wait for whole system to complete before test
- Top down – start from top level components
- Bottom up – start from smaller lower-level modules then integrate
- Incremental – system is built and tested bit by bit
- Interface tests
- Concurrency tests
- API tests

UNIVERSITY OF PLYMOUTH

System Tests / E2E Tests

- Functional Test – check all functional requirements met
- UI test – ease of use for users
- Usability tests – whether a transaction can complete
- Security tests – checks for vulnerabilities
- Regression tests – new features do not introduce defects
- Backup tests

Acceptance Test

- Do this at the end of the module with clients for signoff.

UNIVERSITY OF PLYMOUTH

How should you run tests?

- Run unit tests individually on your own units developed whenever completed.
- Run integration periodically before client meetings as a group to check your units integrate and you keep the business plan in line.
- Run a system test with the client on completed modules during biweekly meetups to check that you have the big picture in sight.
- Run acceptance test at the end of the module to deliver the final product to the client.

UNIVERSITY OF PLYMOUTH

Any Questions?

UNIVERSITY OF PLYMOUTH