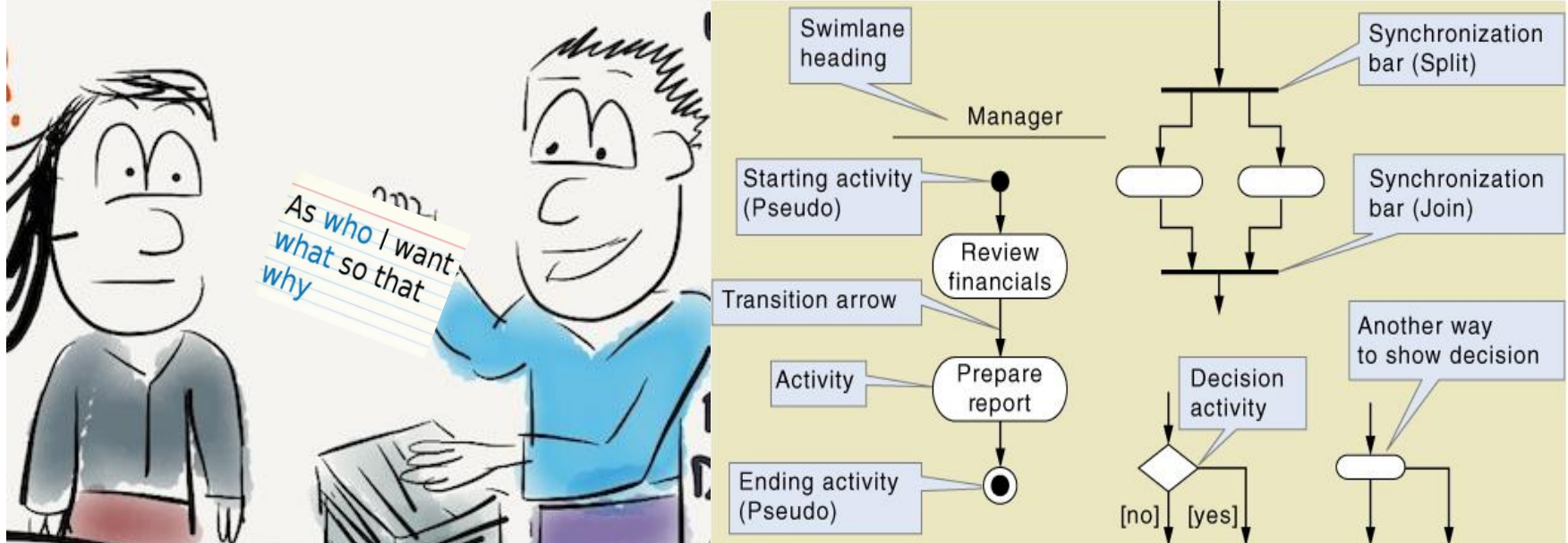# FIT2001 – Systems Development

Seminar 4:  Investigating and documenting system requirements
                    User Stories, Activity Diagrams

Chris Gonsalvez

# Our road map:

- Why modelling?

- Investigating and documenting requirements

  - User Stories

  - Activity Diagrams

- What are Information Systems?

- How do we develop them? Systems Development (SDLC) – key phases

- Development Approaches – Focus on Agile

- Some System Development roles and skills

- Understand the requirements gathering process

- Managing stakeholders

- Requirements gathering techniques

# At the end of this topic you will:

- Explain the value of modelling in systems development;
- Be able to document system requirements using user stories;
- Understand the process for developing suitable user stories for a given scenario;
- Be able to document workflows with Activity diagrams.

# Models - Why do we use them?

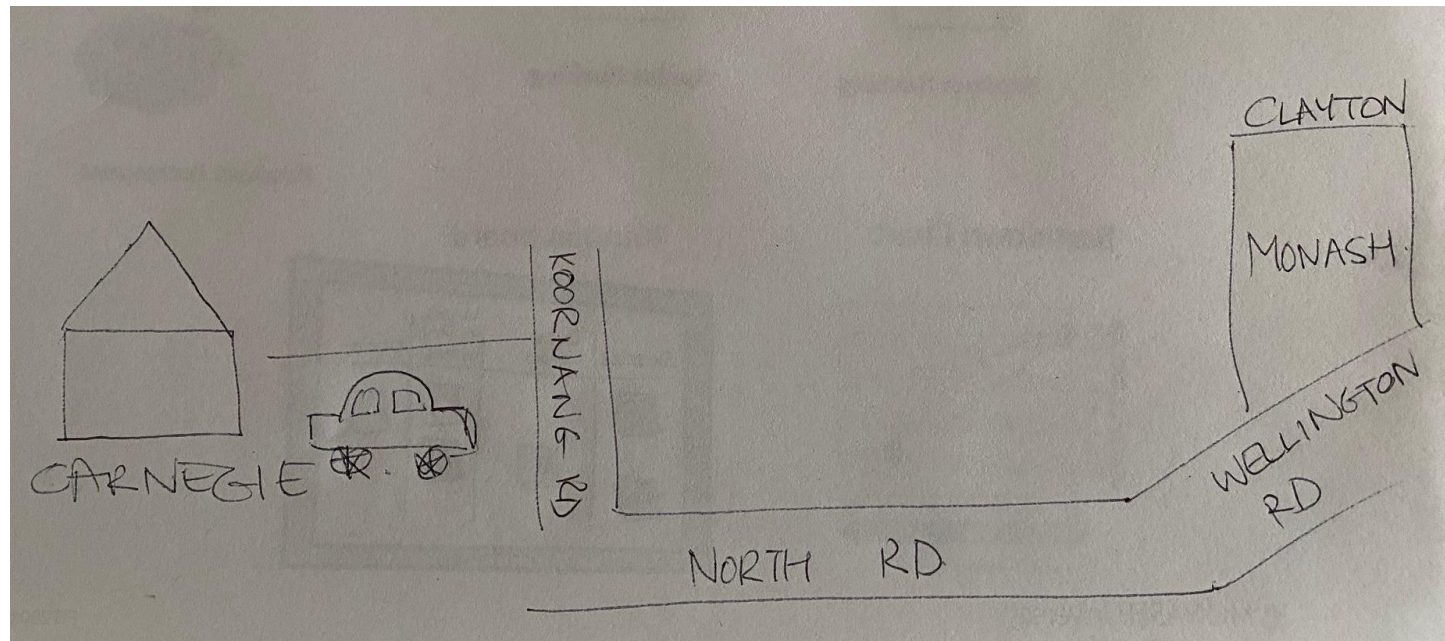- The term 'modelling' has several interpretations

- In its simplest form, the model serves as an abstraction - an approximate representation of the real item that is being built

- It is a simplified picture of complex reality

# Why model ?

- Spend a few seconds drawing how you would normally get to Uni from your home

# **Reasons for modeling** in systems analysis

- Reducing complexity of systems to be built by abstraction

- Communication with other development team members

- Communication with stakeholders/users

- Learning from the modelling process

- Documenting all the details of requirements for future maintenance/enhancement - *represents some key aspects* of the system being built

- *Some examples: Activity diagrams, Class diagrams, Use Case diagrams, Sequence diagrams*

# Gathering requirements in Agile
# User Stories – What are they?

- Short, simple description of a product feature – told from the perspective of the user who wants that feature

- They go into the Product Backlog

As <persona>,
I want <what>
so that <why>.
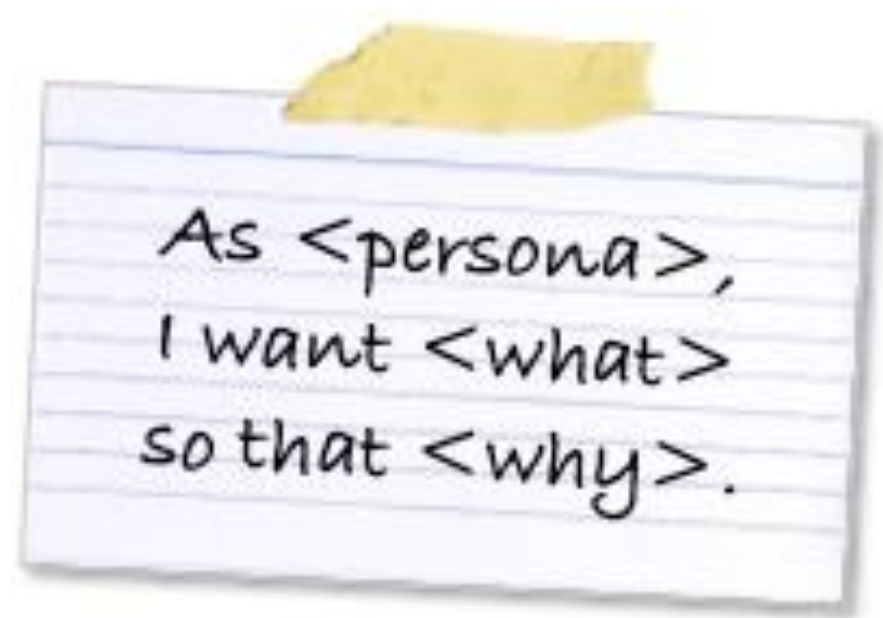
# Examples of User Stories:

- As a student, I want to find my grades messaged to me so that I know my results at the earliest possible time.

- As a book buyer, I want to read reviews of a selected book so that I can make a decision to buy it or not.

- As a parent, I want to see my child's attendance online so that I can take appropriate action if they are not attending

This is high level – they also have acceptance criteria

# **Why User stories**?

- Encourages user communication and collaboration and real-time feedback

- Focus on end user value

- Planning is simplified – if it's too big and you can't estimate, make it smaller

- Avoids locking in design detail too early – focuses on the WHAT - leaves the technical aspects to the developers, testers, etc.

- Users do not need to be trained to understand User stories

- Never out of date … just in time

- Eliminates weighty documentation – create what you need to deliver the story

# Why User stories?

- **Easier to communicate with users – Example:**

   Where does your mind go when I say:

   – Steel body
   – 4 wheels
   – Tyre mounted to each wheel

   *The focus is on System attributes*

   User stories focus on User's goals

   As a lazy man I want to mow my lawn quickly and easy

   As a lazy man I want to feel comfortable while mowing my lawn

# Writing User Stories - The process
# 7.  Story Writing Workshops (Seminar 3)

- An good way to have a shared vision for the product

- Encourage the entire team to participate in the process - includes Product Owner, Development Team, and can include Users and Stakeholders  … Scrum Master usually facilitates
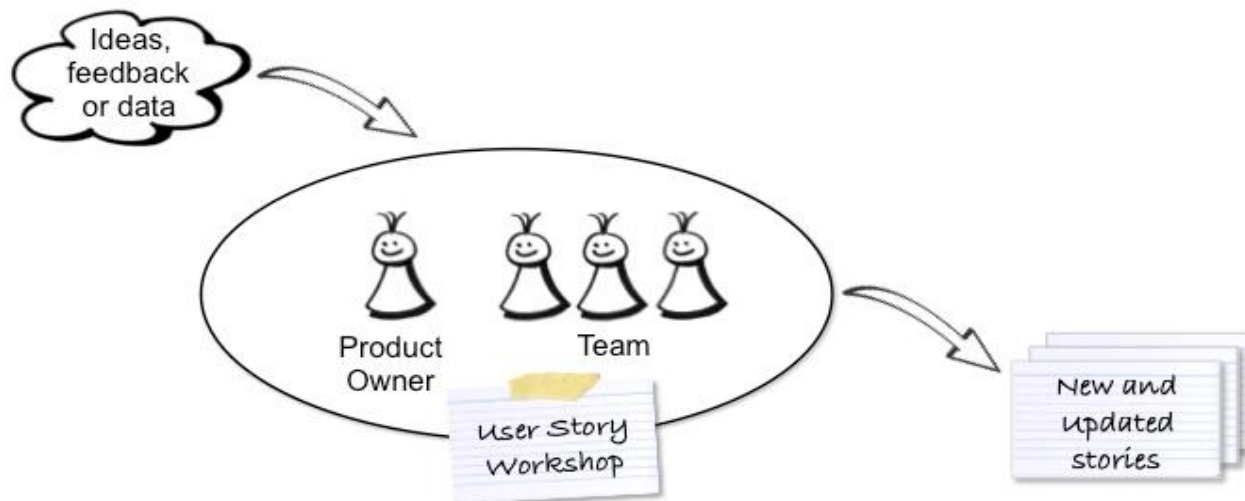
# Running Story Writing Workshops

**Possible steps**

1. Focus the meeting on a single objective
2. Map the big picture – brainstorm, list of things users do, cluster, remove duplicates
3. Add user stories
4. Visualise the relationship between stories - Story mapping
5. Map out releases

- Sticky notes, markers, whiteboard - very visual
- Not too long, very interactive

# How do you write User Stories

## … The 3 Cs  by Ron Jefferies

### CARD

Traditionally written on index cards – short, concise

Annotated with notes, models, prototypes

### CONVERSATION

Story details come out of conversations with the Product Owner (User)

### CONFIRMATION

What is going to make the User accept the story as completed?

# How do you write User Stories?
# … Use a template as a starting point

1. Define your **end user**
2. Specify what **they want**
3. Describe **the benefit**
4. Add **acceptance criteria**

Copyright © 2015 Knowledge Train Limited

… Don't get hung up on the words and the format  - the key is to understand what the user wants

"As a <user role>, I <want/need/can/ etc> <goal> so that <reason>."

# How do you write User Stories? … Identify the user type

- Avoid the generic role "User"

- Focus on the personas who interact with the system or who realize some value or benefit from the system

  - Who is this user? What motivates them? Who is an example of such a user?

Rita the Reseller

Rhonda the Receptionist

Amy the Assistant

Esteban the Executive

Chuck the Call Center

# How do you write User Stories?
# … Ask questions

- What else might users of the system have done?

- What could go wrong, and what would the user have to do to recover?

- What might other types of users do to reach their goals?

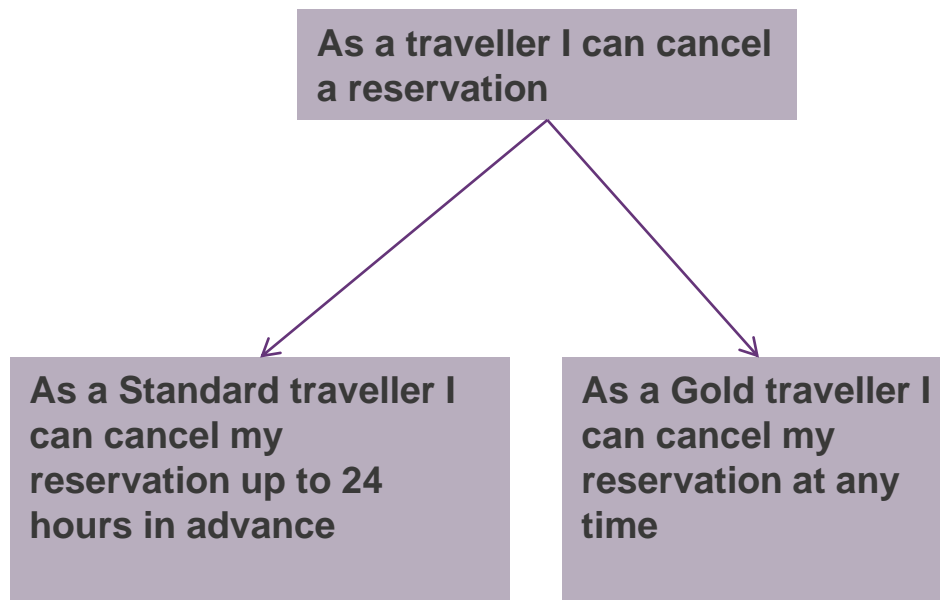- Wouldn't it be cool if... ?

# How do you write User Stories?
# … get the size right

- Should be small enough for the development team to create in a short time period.

- Should be big enough to represent business value in its own right -- it might build on something that has been done before

- Should be able to be delivered in its own right -- you might not want to do so but if you needed to you could.

- User stories that are too big, are harder to understand, estimate, and implement successfully.

# How do you write User Stories? … Story too large?

- EPIC - a LARGE story that is split into smaller user stories that share a strategic objective

- Product Backlog items tend to start as epics that are broken into smaller user stories during Sprint planning

**As a traveller I can cancel a reservation**

**As a Standard traveller I can cancel my reservation up to 24 hours in advance**

**As a Gold traveller I can cancel my reservation at any time**

- An epic will typically be delivered over several sprints.

# How do you write User Stories?
# … adding the Acceptance criteria

As a credit card holder, I want to view my statement balance, so that I can pay the balance due

- Display statement balance upon authentication
- Display Total Balance
- Show "Payment Due Date" and "Minimum Payment Due"
- Display Error message if service not responding/ timeout

Acceptance criteria - a set of predefined requirements that must be met in order to mark a user story complete.

- Reduces ambiguity

- Prevents miscommunication

- Clearly defines what you need to test to meet the user's requirements

# Example 1:

As a purchaser on the website

I want the ability to pay with a credit card

So that I can confirm my purchase immediately

***Acceptance criteria:***

☐ Accept Visa, Mastercard, Amex

☐ Accept valid credit cards – valid cc#, valid expiry date, valid cvv no.

☐ Generate purchase payment confirmed message

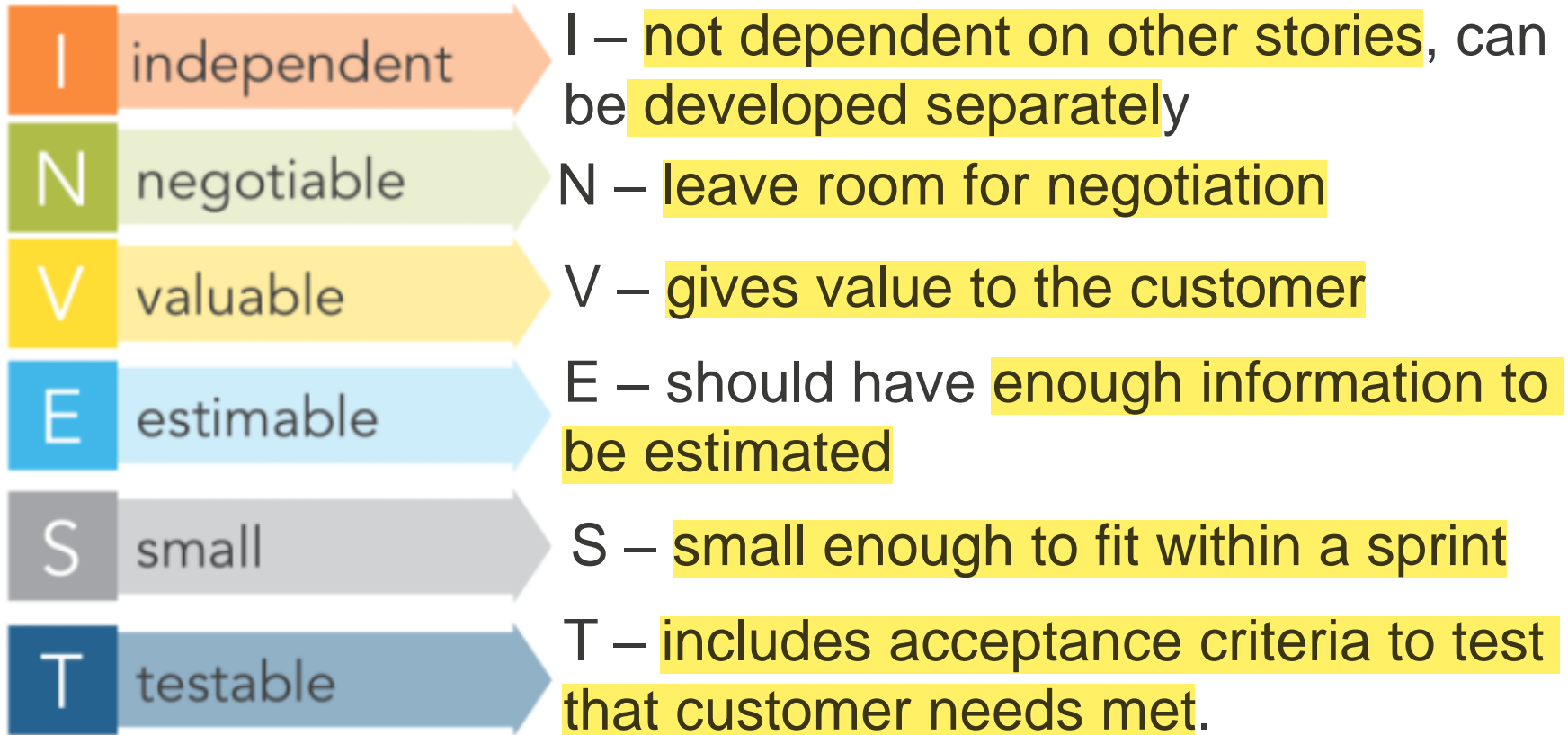☐ Generate purchase payment failed message

# Example 2:

As an online customer

I want to be able to register online

So that I can start shopping online

## *Acceptance criteria:*

☐  Customer must be able to submit registration form online

☐  The email used must not be a previously used email

☐  Customer must complete all mandatory fields on form

☐  Customer must receive a confirmation email with link after valid submission

☐  Registration must be confirmed after they click the link

# How good is your User Story?   INVEST



I – not dependent on other stories, can be developed separately

N – leave room for negotiation

V – gives value to the customer

E – should have enough information to be estimated

S – small enough to fit within a sprint

T – includes acceptance criteria to test that customer needs met.

Ref: Bill Wake – Extreme Programming Explored

# User Story - Common **mistakes**

- Too formal / too much detail

  – could result in skipping the conversation

- Technical tasks impersonating user stories

  – Does this represent what the user wants?

- Skipping the conversation

  – Risk moving in the wrong direction, overlooking specific customer needs

READ:  Writing a great user story:  https://help.rallydev.com/writing-great-user-story

# Writing User Stories

- If Monash was developing a new Web Enrolment system – write 1 User Story for the 'Student' user and 1 User Story for the 'Lecturer' user

"As a <user role>,
I <want/need/can/
etc> <goal>
so that <reason>."

- Did you follow the template
- Did you remember the

conditions of satisfaction (the acceptance criteria)

# Story Mapping

- Helps arrange user stories into a useful model for understanding the functionality of a system

- Helps identifying holes and omissions in your backlog, and effectively plan holistic releases that deliver value to the business

- Walk through the activities and ask questions such as:

  - What will the user most likely want to do next?
  - What mistakes could a user make here?
  - What could confuse a user at this point?
  - What additional information could a user need?

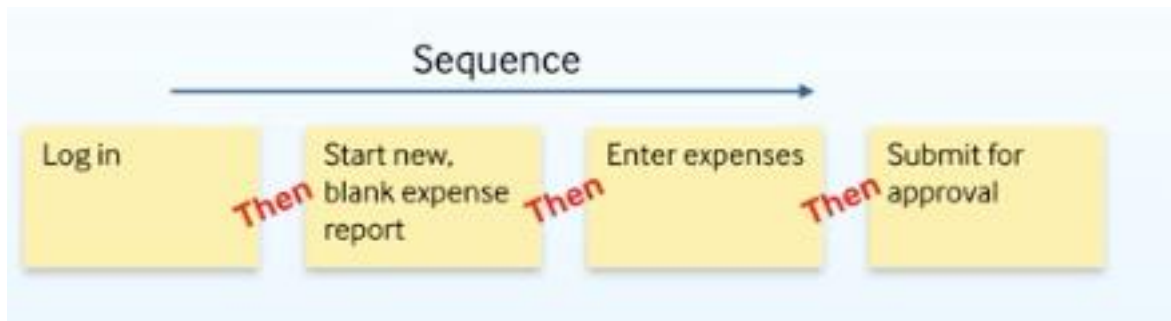- *If multiple personas ask questions for each persona*

# Story Mapping

- Example – Develop software to allow company employees to submit their expense reports and be reimbursed
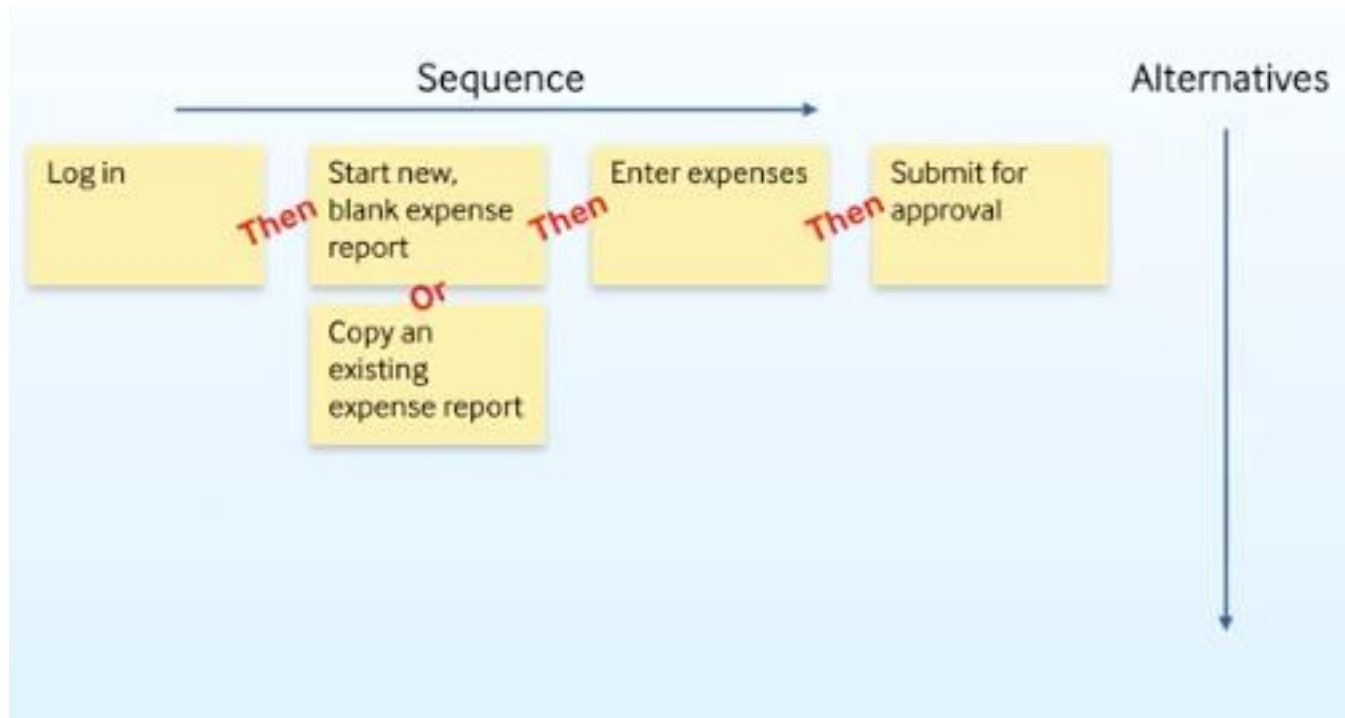
Ref: Michael Cohn – Better User Stories

- Two dimensions
    – Sequence
    – Alternatives

1. We start with the sequence of activities in the process – ask Questions

# Story Mapping

2. We then look at possible alternatives for activities

# Story Mapping

3. More alternatives identified, a forgotten process added – the higher the value the higher up the story is placed on the alternative axis.

# Release planning

- A Release is a set of functionality that makes sense to deliver to users at the same time – can include stories from different Epics

- Create horizontal swim lanes for each release – move stories up or down to form collections that would be the most important to build
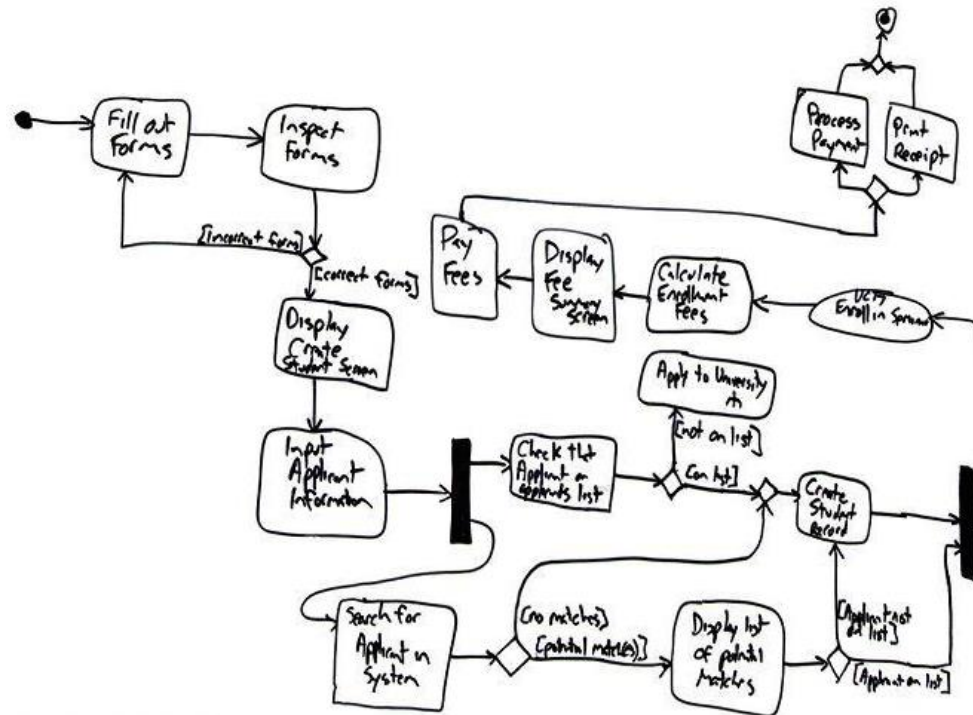
MONASH University
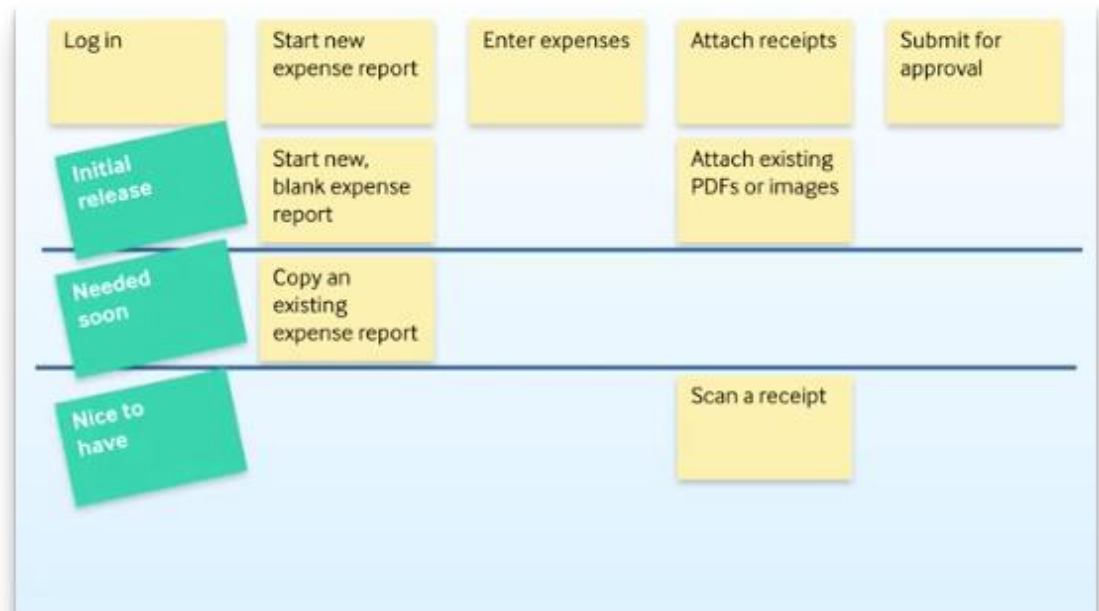
# Release planning

- A Release can include multiple sprints

- Release 1 – 4 user stories make sense to be given to users
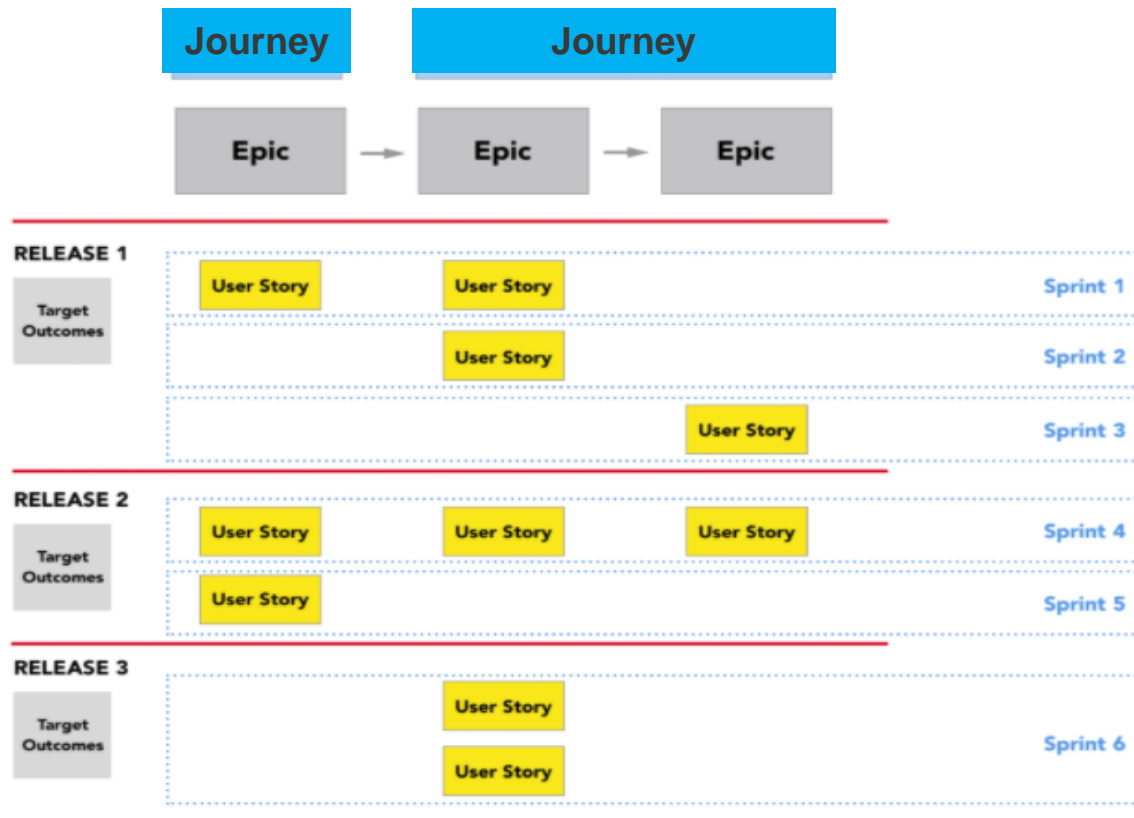
# Activity diagrams

- "Activity diagrams are a technique to describe procedural logic, business processes, and work flows" - M. Fowler

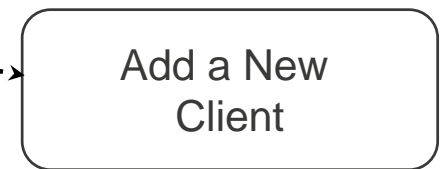- They add meaning/detail to our user stories (if required)



Copyright 2005 Scott W. Ambler

# Activity diagram symbols.1

- Activity / Action
  - rectangle with rounded corners
  - meaningful name

- Control flows
  - arrows with open arrowheads

- Notes

Add a New Client

Assign Staff Contact

{Client logs out of the system.}
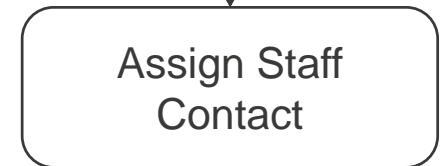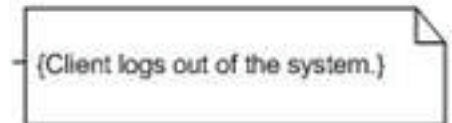
# Activity diagram symbols.2

- Initial node
  - black circle

- Decision nodes (and merge nodes)
  - Diamond

- Guard conditions
  - in square brackets

- Final node
  - black circle in white circle

Add a New Client

Assign Staff Contact

[no campaign to add]

[campaign to add]

Add New Campaign
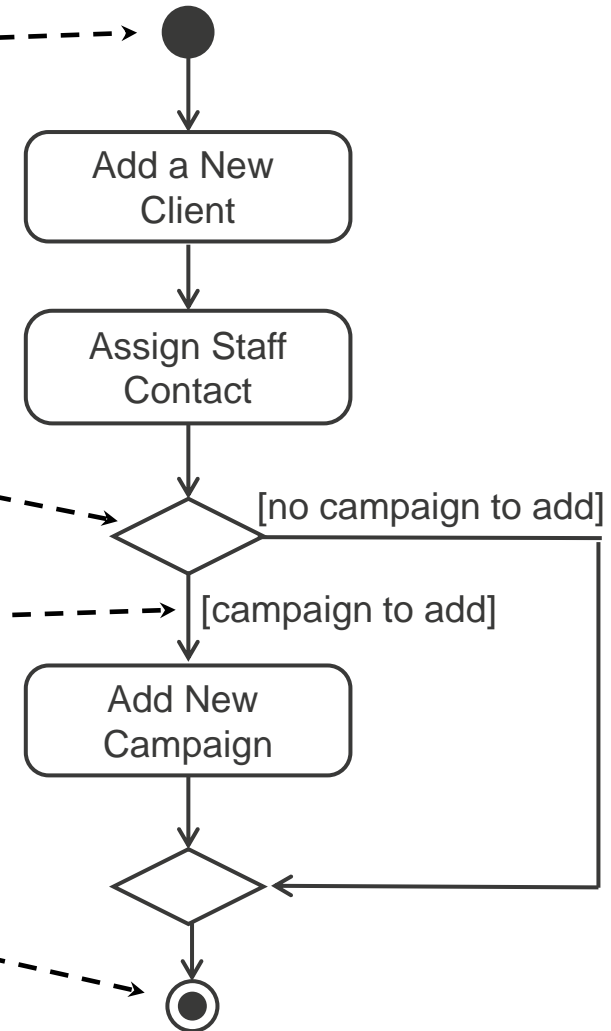
# Activity diagram symbols.3

- Split/Fork nodes and Join nodes
  - thick bar
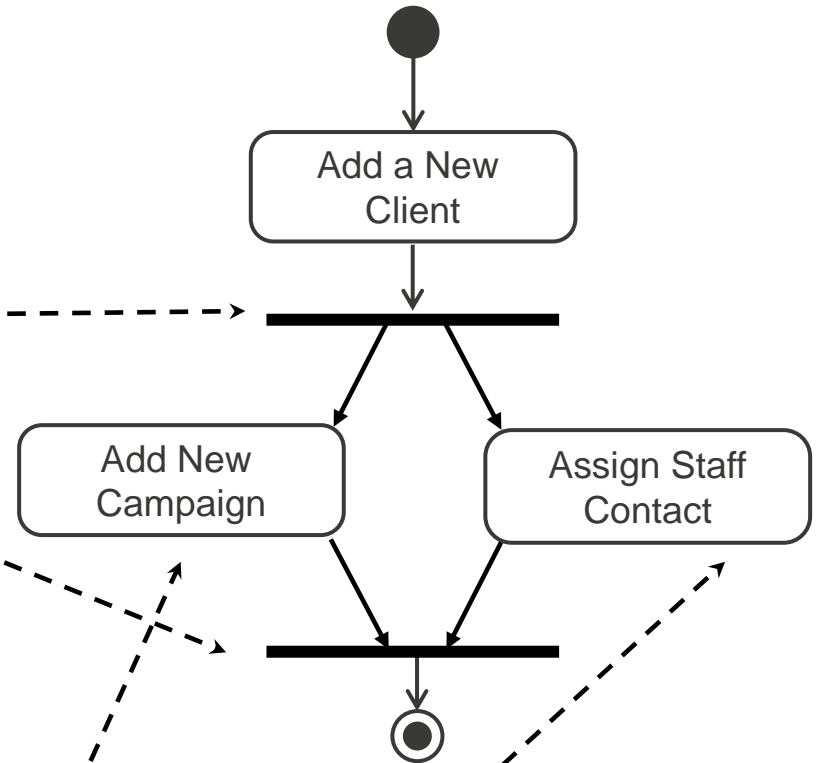
» Actions carried out in parallel

**Add a New Client**

**Add New Campaign**

**Assign Staff Contact**

# Activity diagram symbols.4

- Object flows
  - open arrow

- Objects
  - rectangle
  - optionally shows the state of the object in square brackets

:Campaign
[Active]

Record completion
of a campaign

:Campaign
[Completed]
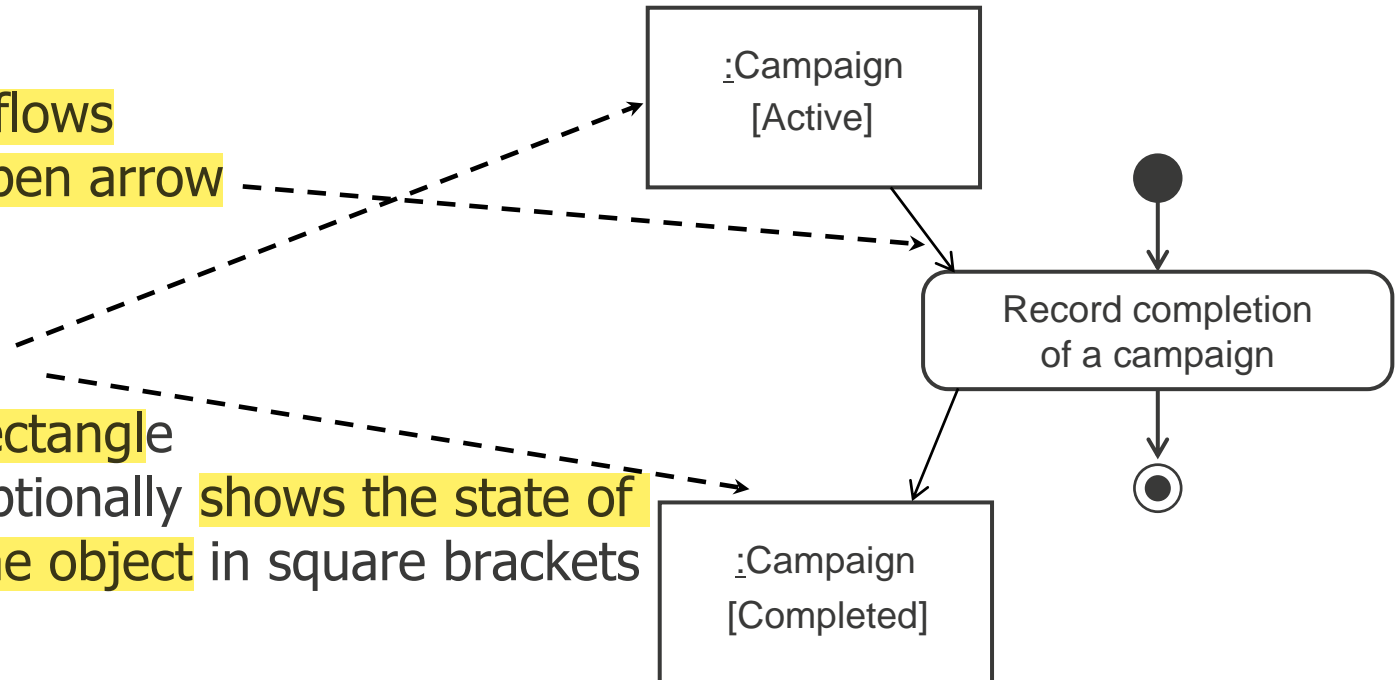
# Activity diagram symbols.5

- Activity Partitions (Swimlanes)
  - vertical columns
  - labelled with the person, organisation, department or system responsible for the activities in that column

| Campaign Manager | Accountant | Client |
|---|---|---|

- Record Completion of a campaign
- Issue invoice
- Pay invoice
- Record client payment

© **Bennett, McRobb and Farmer 2005**
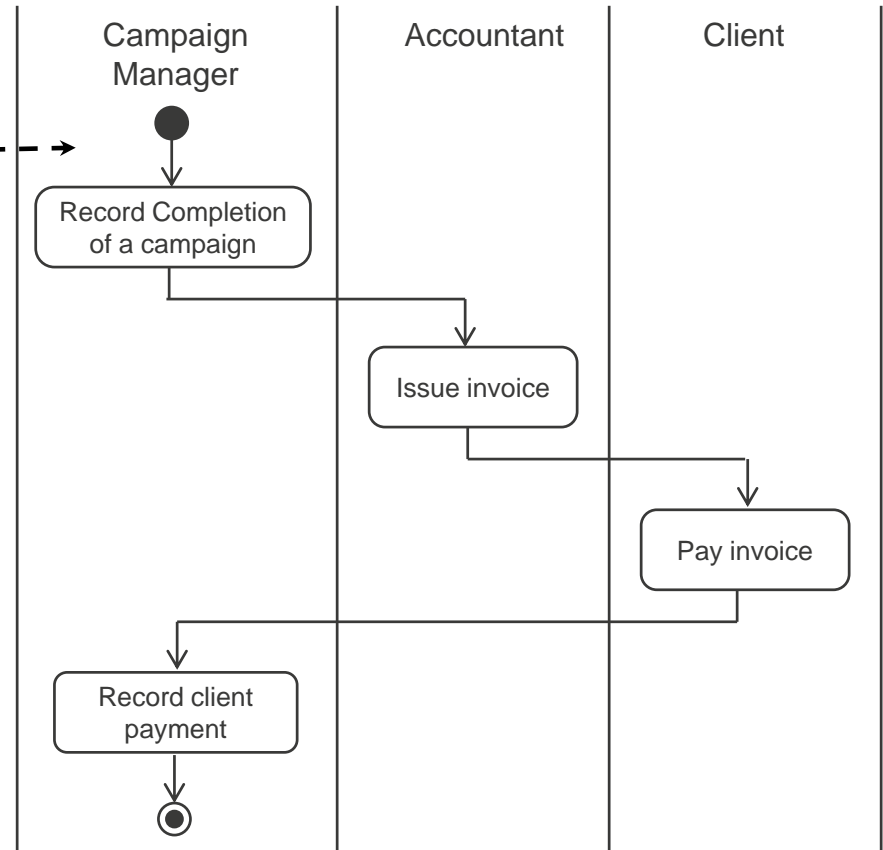
# Scenario for Activity diagram discussion

- *When a new client has been accepted by our marketing firm, we assign them a staff contact to look after them.  If they have requested a marketing campaign, we add a new campaign. We then assign staff resources to the campaign as required.*

# Drawing Activity diagrams.1

- Identify actions

  - What happens when a new client
    is added to the system

    - Add a New Client
    - Assign Staff Contact
    - Add New Campaign
    - Assign Staff to Campaign

- Organise the actions in sequential order with flows

```
  ●
  │
  ▼
┌──────────────┐
│  Add a New   │
│    Client    │
└──────────────┘
  │
  ▼
┌──────────────┐
│ Assign Staff │
│   Contact    │
└──────────────┘
  │
  ▼
┌──────────────┐
│   Add New    │
│   Campaign   │
└──────────────┘
  │
  ▼
┌──────────────┐
│ Assign Staff │
│ to Campaign  │
└──────────────┘
  │
  ▼
  ◉
```
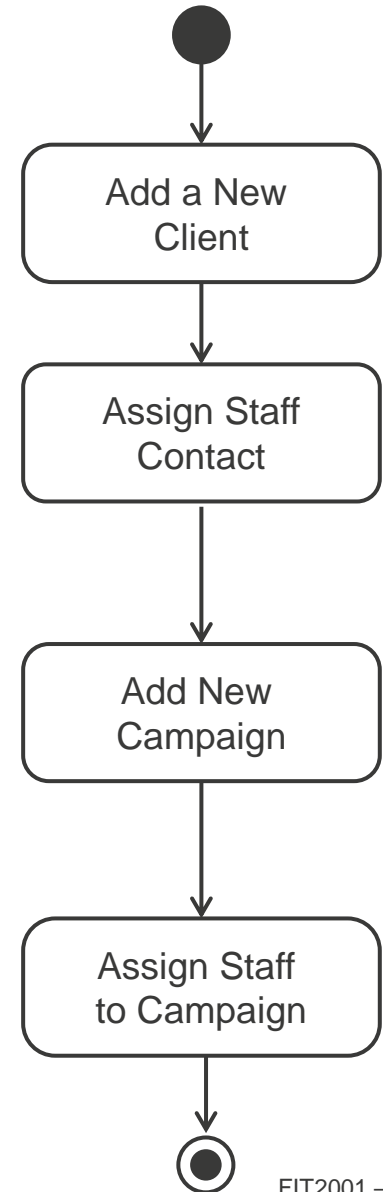
# Drawing Activity diagrams.2

- Identify any alternative flows and the conditions on them
  - sometimes there is a new campaign to add for a new client, sometimes not
  - sometimes they will want to assign staff to the campaign, sometimes not

- Add decision and merge nodes, flows and guard conditions to the diagram

Add a New Client

Assign Staff Contact

[no campaign to add]

[campaign to add]

Add New Campaign

[no resources required]

[resources required]

Assign Staff to Campaign

© **Bennett, McRobb and Farmer 2005**
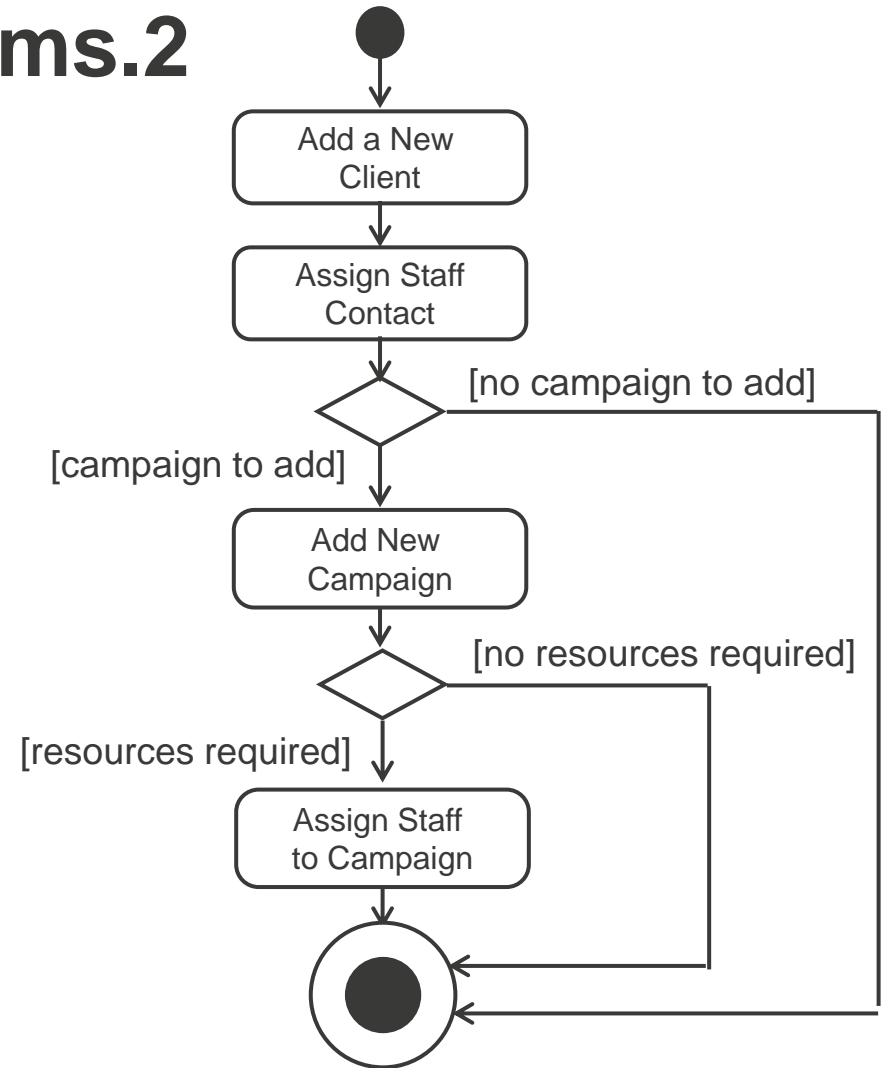
# Drawing Activity diagrams.3

- Identify any actions that are carried out in parallel
  - there are none in this scenario, but what if we said that if resources are required, we assign staff and allocate funds

- Add fork and join nodes and flows to the diagram

# Drawing Activity diagrams.4

- Identify any processes that are repeated
  - they will want to assign staff to the campaign until there are no more staff required

- Add decision and merge nodes, flows and guard conditions to the diagram

Add a New Client

Assign Staff Contact

[no campaign to add]

[campaign to add]

Add New Campaign

[no resources required]

[resources required]

Assign staff to campaign

Allocate additional funding

[more resources required]

[no more resources required]
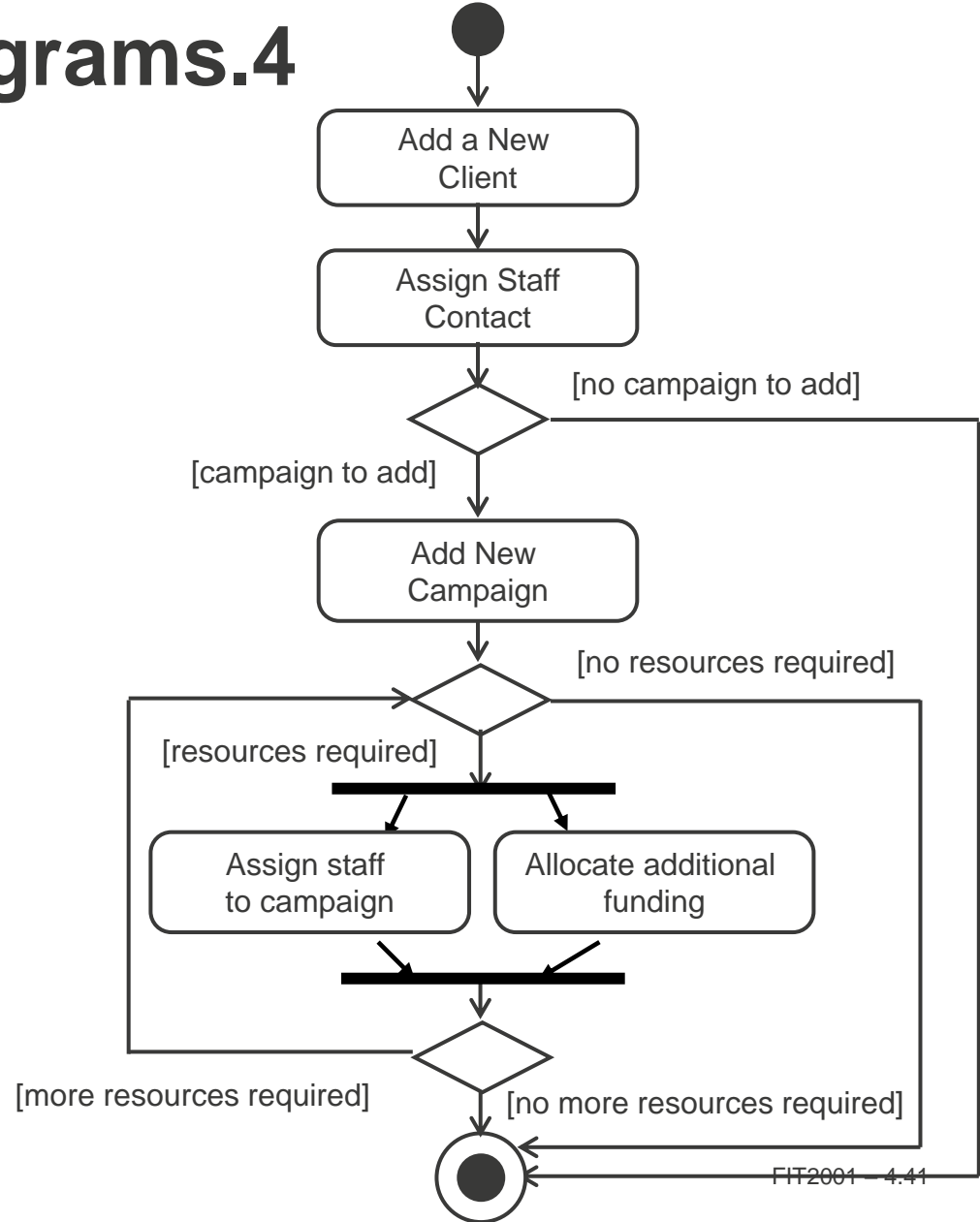
# Drawing Activity diagrams.5

- Are all the activities carried out by the same person, organisation or department?

- If not, then add swimlanes to show the responsibilities

- Name the swimlanes

- Show each activity in the appropriate swimlane

| Administrator | Campaign Manager |
|---|---|

Add a New
Client

Assign Staff
Contact

Add New
Campaign

[no resources required]

[campaign
to add]

[no campaign
to add]

[resources required]

Assign Staff
to Campaign

[more resources
required]

[no more resources required]
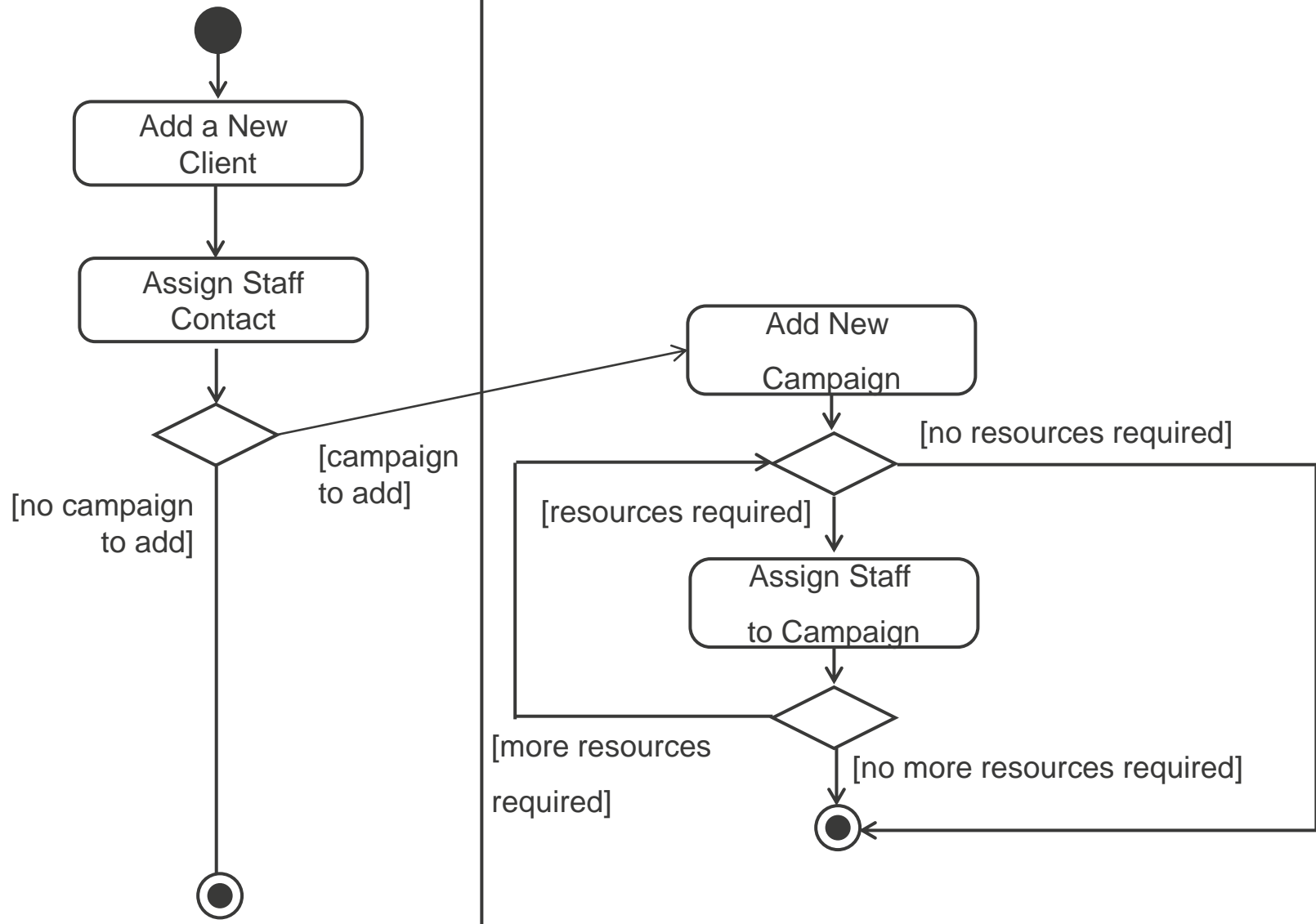
# Drawing Activity diagrams.6

- Are there any object flows and objects to show?
  - these can be documents that are created or updated in a business activity diagram
  - these can be object instances that change state in an operation or a use case

- Add the object flows and objects

| Administrator | Campaign Manager |

Add a New Client

:Client [New]

Assign Staff Contact

Add New Campaign

:Campaign [Commissioned]

[no resources required]

[campaign to add]

[no campaign to add]

[resources required]

Assign Staff to Campaign

[more resources required]

[no more resources required]

| Administrator | Campaign Manager |
|---|---|

Add a New Client

:Client [New]

A

Assign Staff Contact

B

Add New Campaign

:Campaign [Commissioned]

[no campaign to add]

C

D [campaign to add]

E

[no resources required]

F

[resources required]

G

Assign Staff to Campaign

H

[more resources required]

I [no more resources required]

# ON THE SPOT COURIER SERVICES

Bill Wiley – Pick-up and Delivery function

When Bill Wiley receives a request for pickup, he checks to see if the pick up is valid, and rejects it if it isn't valid. If it is valid, he enters the pickup information on a form and processes the payment. If the payment is approved, he contacts his courier staff with the pick up information, otherwise he rejects the request. When the courier picks up the package they confirm pickup. If the package is not available for pickup, the request is rejected.

**Draw an activity diagram for the Pick-up and Delivery function for On the Spot Courier Services**

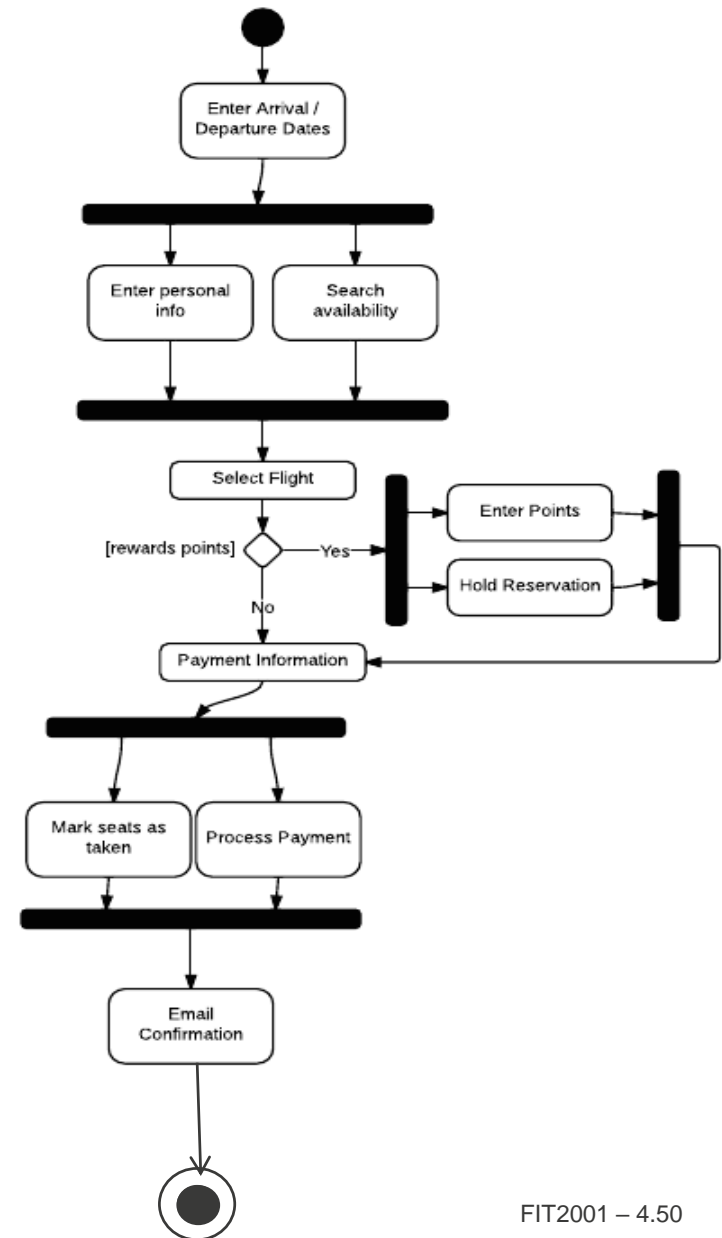# Pick-up and Delivery function: Activity Diagram
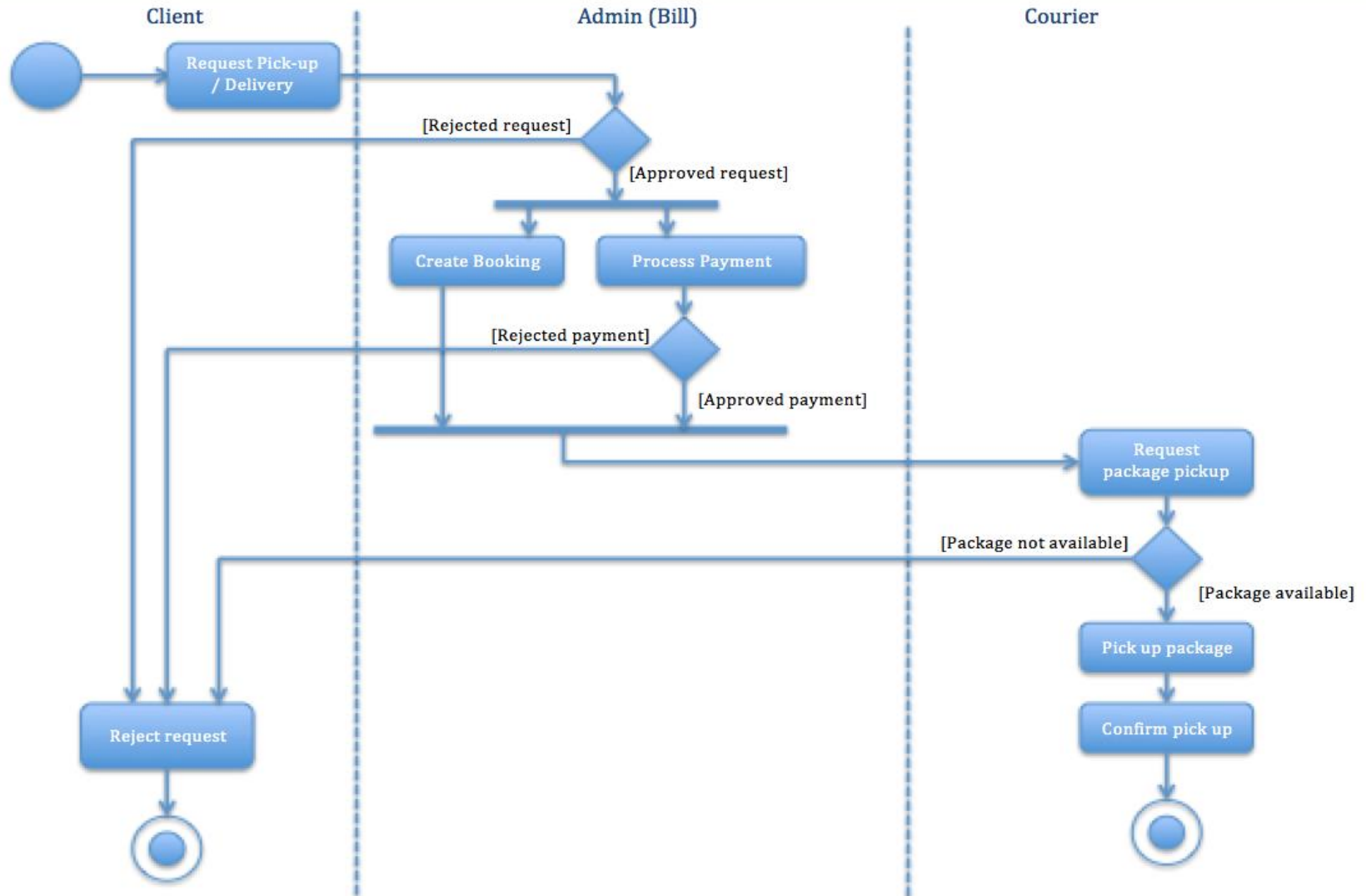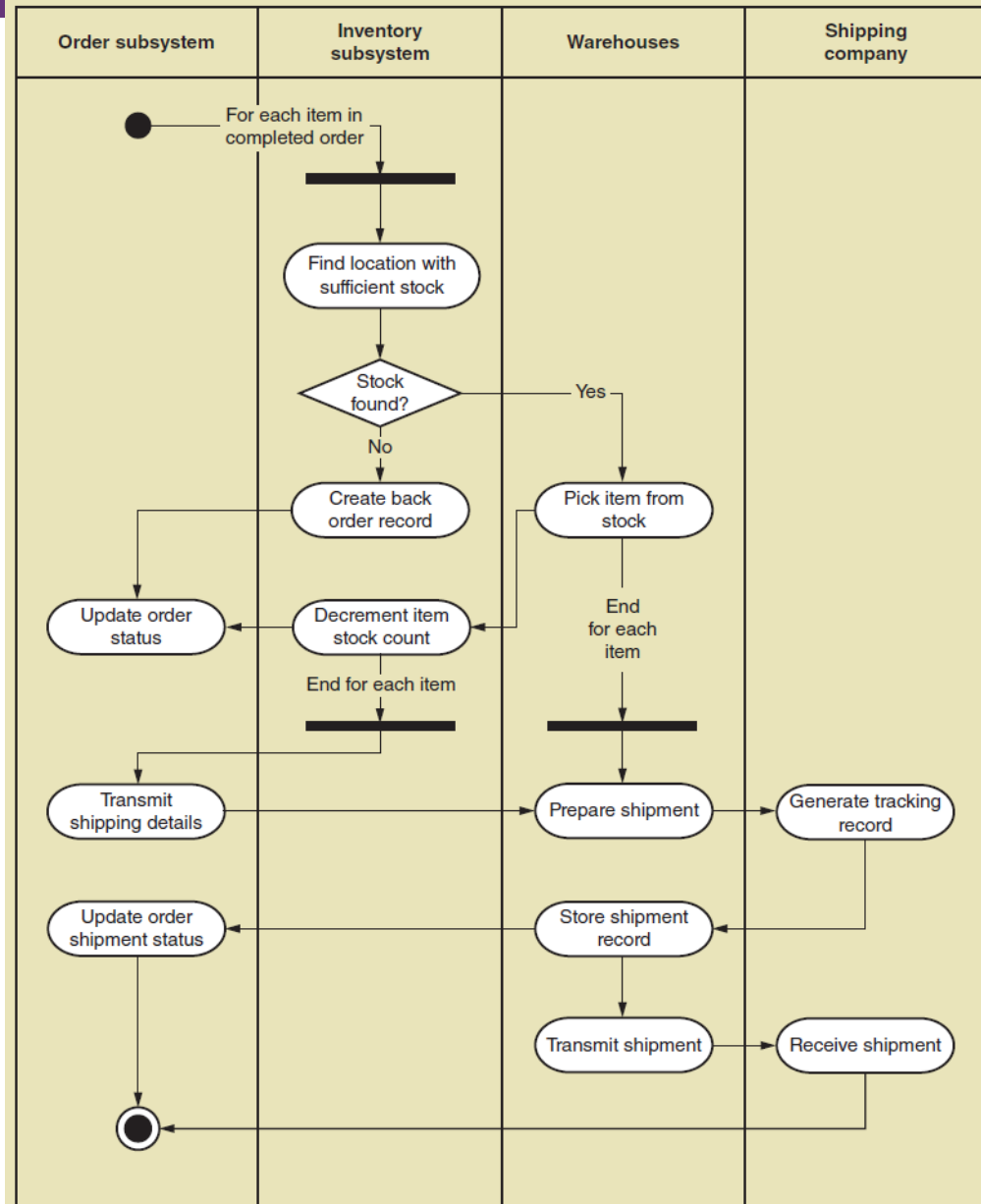
# Additional examples to help you …

# Example: Airline Reservation

This example shows the process of a reserving a flight. First, you enter the dates. Once you submit your desired flight plan, you can enter your personal information and at the same time the system could be searching availability. The system flow then joins back into one and you can select the specific flight on the dates you want to fly. This activity diagram gives you two different paths dependent on whether you are using reward points. After entering payment information, the system performs two processes at the same time and then sends out a confirmation email.

# Example from Text: Activity Diagram for Order Fulfillment



**Order Fulfillment**

| Order subsystem | Inventory subsystem | Warehouses | Shipping company |
|---|---|---|---|

For each item in completed order

- Find location with sufficient stock
- Stock found? — Yes / No
- Create back order record
- Pick item from stock
- Update order status
- Decrement item stock count
- End for each item
- Transmit shipping details
- Prepare shipment
- Generate tracking record
- Update order shipment status
- Store shipment record
- Transmit shipment
- Receive shipment

# Workshop Preparation

- Watch Seminar 4

- Review any resources

- Assignment 1 Interview preparation

  - Meet with your team regularly to prepare to complete all the requirements for Assignment 1
  - Be prepared for your first interview with your client.

# Thanks for watching
# See you next week

MONASH University

# Resources:

**Prescribed text:**

– Satzinger, J. W., Jackson, R.B., and Burd, S.D.(2016) Systems Analysis and Design in a Changing World, 7th Edition, Cengage Learning, Chapters 2 (pp. 58-63) & Chapter 3 (pp. 71-73)

**Resources:**

- User Stories
  Videos - http://vimeo.com/43601248
  http://www.agilenutshell.com/episodes/2-userstories

  Article - https://help.rallydev.com/writing-great-user-story