

# Managing Requirements through to Service Specification

Hilary Dexter

Distributed Learning

The University of Manchester

# Overview

- Requirements management:
  - Where the knowledge lies
  - Addressing the right problem
  - Eliciting requirements
  - Managing traceability
  - Managing change
- Use Cases and their workflows/processes
- A service development process
- Mapping Use Cases to components and services
- Supporting the developer

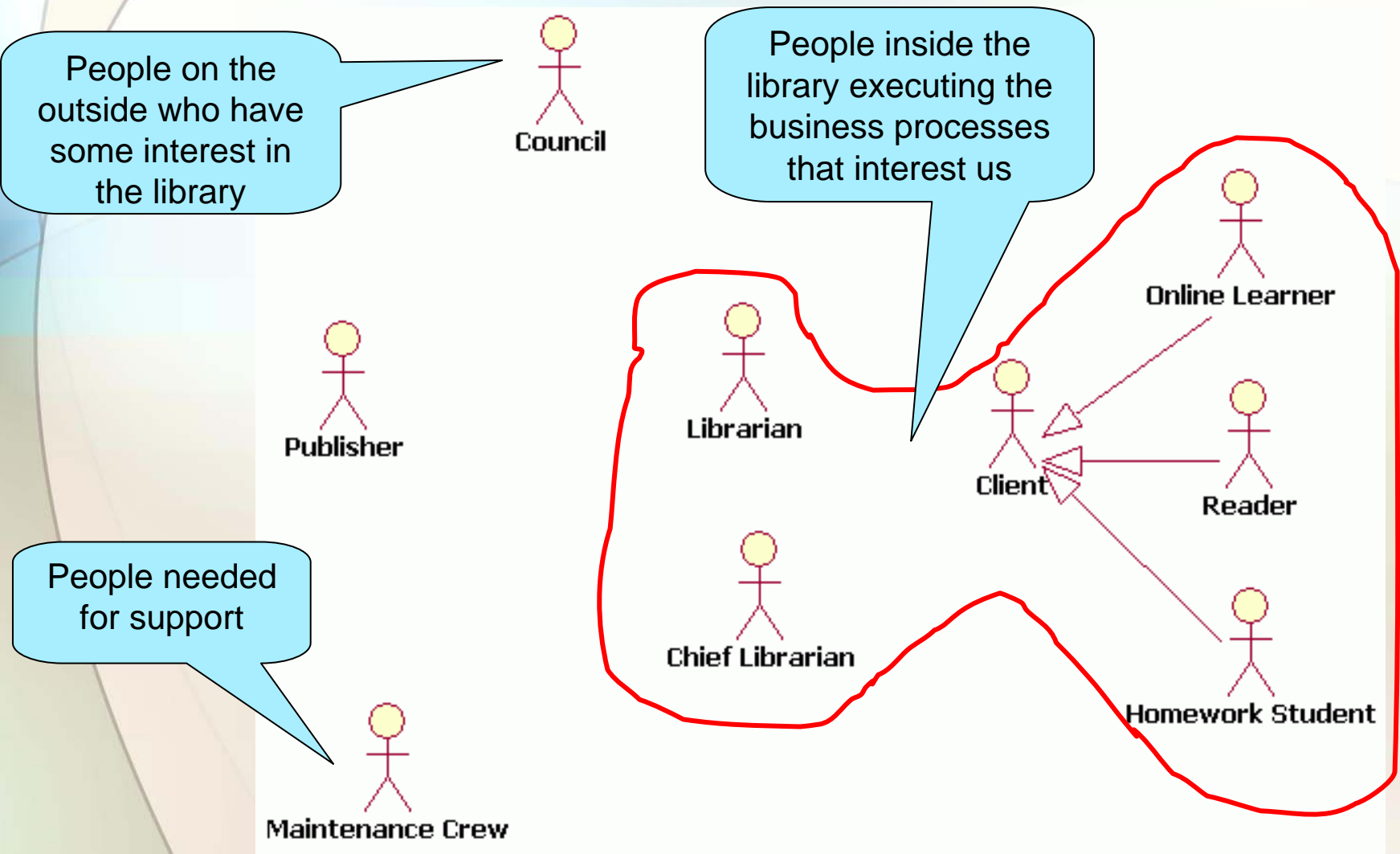
# 5 Key Questions for the domain experts

- Where am I?
- Who is here?
- How do we all carry out our responsibilities?
- What things are here?
- What supports our work?

# Where am I and who is here?

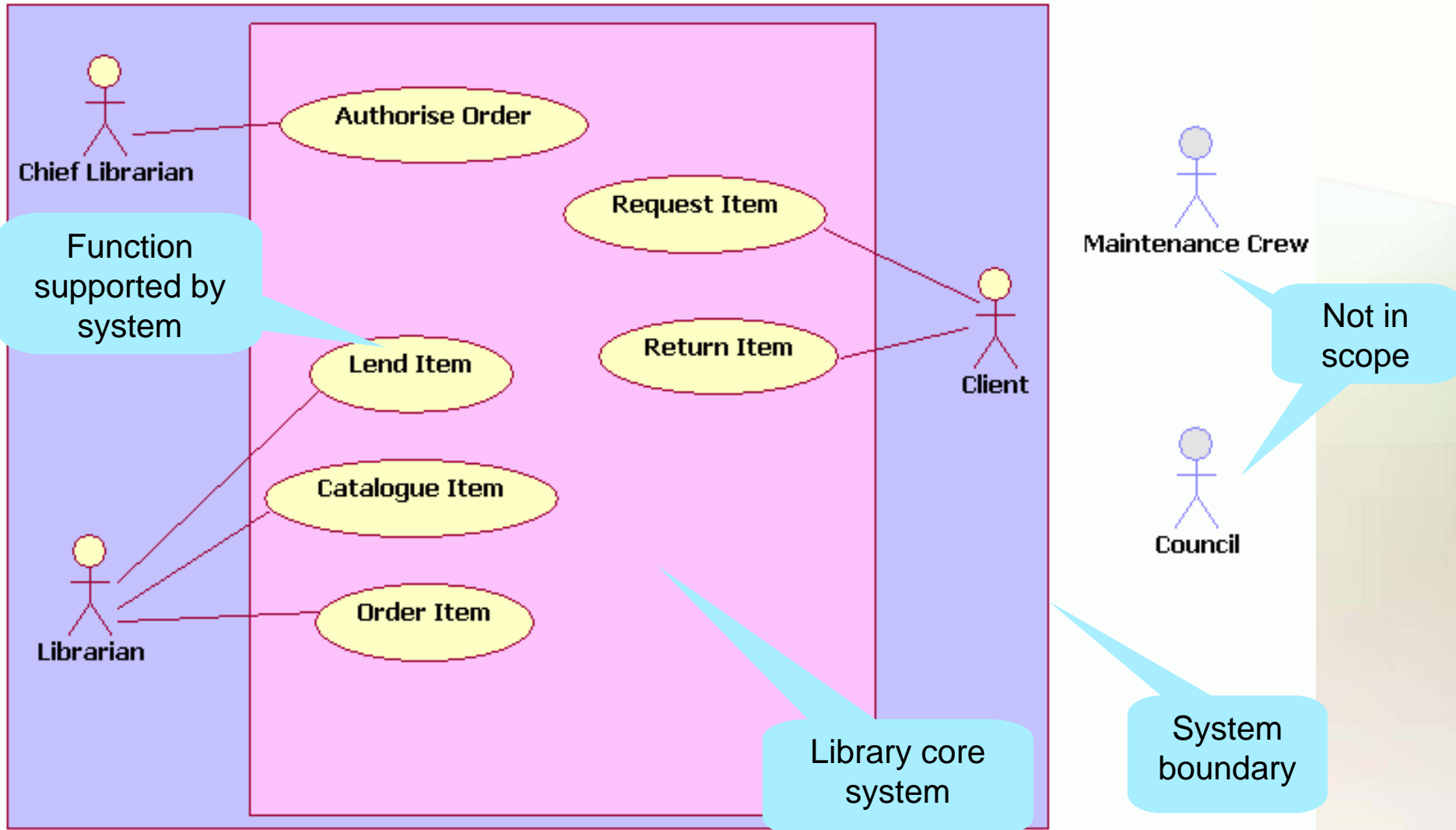
The System: A Public Library (core business is lending items to clients)

The Actors: People (roles) who are involved in the business of the library



# Where am I and who is here?

Use Case Diagram: Roles, functions and system boundary



# Case Study: Supporting Practitioner Tutorials

## ***Supporting Practitioner Tutorials***

(Based on a validation scenario of the JISC HORUS project

<http://www.jisc.ac.uk/index.cfm?name=delethorus> )

Supporting tutorials given by practitioners is a key business process. This process is to be supported by services from the “Teaching and Learning Services Framework” (TLSF). The service to be designed and developed is called “Teaching Event Manager” (TEM) and it may call upon utility services in the TLSF to carry out its functions. In this workshop you will analyse the service requirements to support the following scenario:

A practitioner regularly gives a tutorial on the Clinical Psychology Programme. This scenario concerns how she is recruited to give it, scheduled, briefed and reminded. She uploads her presentation as a learning resource on the programme website, and so learners can consult it afterwards. She rates the resources and processes provided for her, and receives feedback on the students’ reactions to her tutorial... (see handout)

# Abstracting Use Cases from Usage Narrative

## Usage Narrative and Use Cases (UCs)

Usage Narrative

### Linking from the PBL case to clinical ILOs!

Catherine is an MRI student in year three, NME module, attached to a firm. According to Medea the PBL case she will be doing tomorrow is "A sweet life". She has an hour to kill and is planning to go on to the ward to see if she can gain some clinical experience. Medea tells her that "A sweet life" is related to diabetes/endocrine disease and takes her into the ILO browser, where she can look at the skills she is expected to acquire and experiences she should obtain. Because Medea knows where she is in the PBL cycle, it takes her directly to the correct part of NME, which shows her she should obtain clinical experience in diabetes and hypoglycaemia and develop two specific skills: near patient glucose testing and eye examination. They are identified by icons as a lab skill and examination skill respectively.

The screen gives her some extra information about her clinical learning in this part of the NME module. It shows her some upcoming clinics and teaching sessions where she can learn about diabetes and endocrine disease. It shows how her current levels of skill and experience match up to the expected level for her stage of training, and how she compares with her peer group. She sees that her target for near patient glucose testing is level 6 on the competence ladder and she has so far only reached level 5, so to decide what to do about it, she clicks to go to the "ILO: near patient glucose measurement" homepage.

Usage Narrative may be written in free text as above or in a structured Use Case template (see example: Portfolio learning from placement experiences)

**Comment [m1]:**  
High Level UC (Business UC) "Link PBL case to clinical ILOs"

**Comment [m2]:**  
UC Find skills and experiences for current user

**Comment [m3]:**  
UC Display place in module content for place in PBL cycle

**Comment [m4]:**  
Non-functional requirement for UI - use of icons

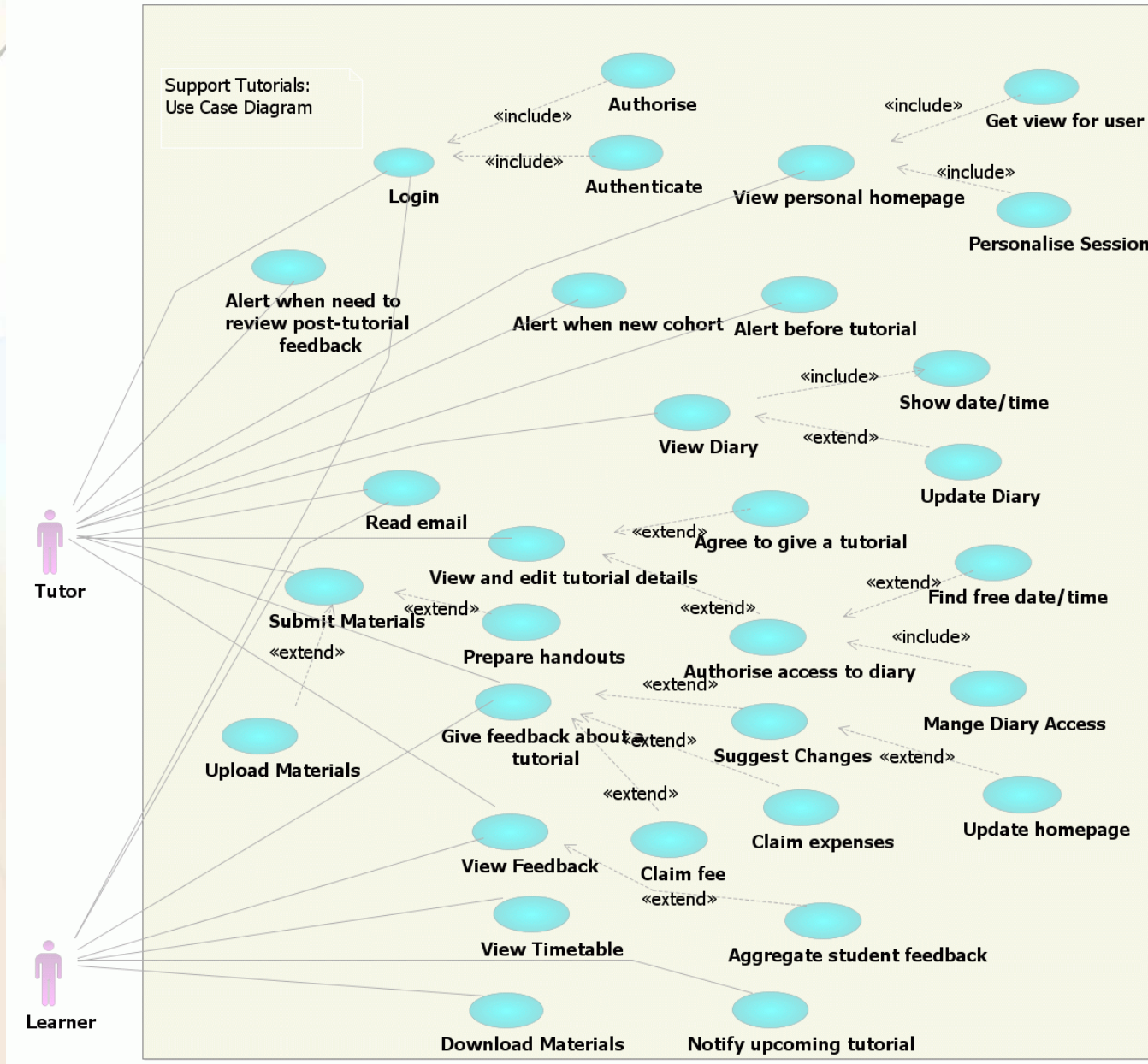
**Comment [m5]:**  
UC find relevant learning opportunities for place in module

**Comment [m6]:**  
UC Display current levels of skills and experience

**Comment [m7]:**  
UC Compare current user with peer group

**Comment [m8]:**  
UC Plan learning opportunities

# Support Tutorials: Use Case Diagram



# Structuring a Use Case Document

1. Brief Description  
Conveys the role and purpose of the use case.
2. Basic Flow of Events  
What the actor does and what the system does in response - a dialog between the actor and the system.
3. Alternative Flows  
Represent alternative behavior usually due to exceptions that occur in the main flow.
4. Preconditions  
The state of the system that must be present prior to a use case being performed.
5. Postconditions  
A list of possible states the system can be in immediately after a use case has finished.
6. Extension points  
Show where extending use cases are allowed to add functionality.
7. Special Requirements  
Typically a nonfunctional requirement that is specific to a use case.
8. Additional Information  
Include, or provide references to, any additional information required to clarify the use case.

# What do we mean by 'Business Process'?

A business process is "a structured, measured set of activities designed to produce a specified output for a particular customer or market"

Davenport (1993).

In our case, an example of a library's business process would be "lend a library book to a reader", or, more generally, "lend an item to a client"

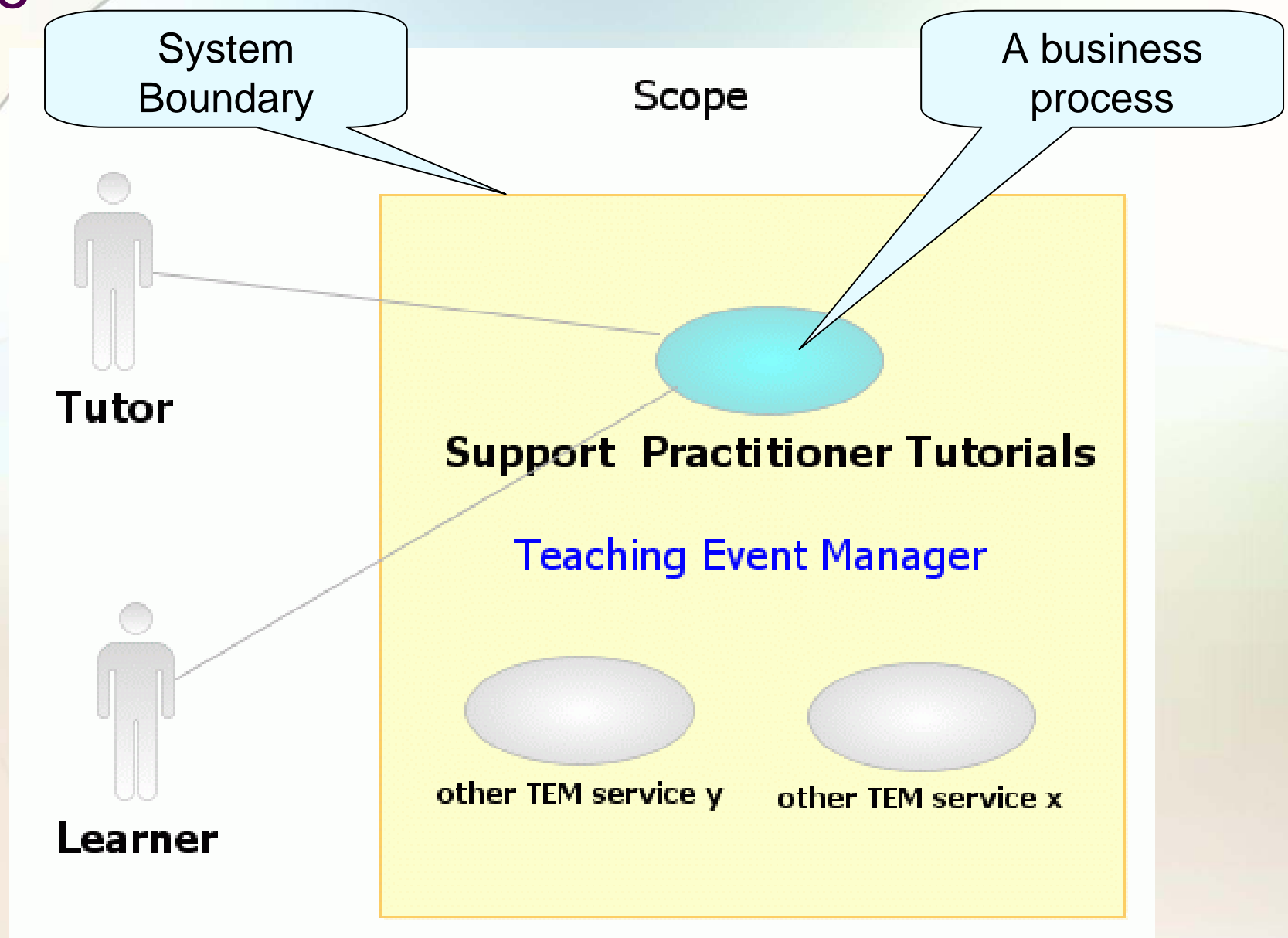
**e.g. An HEI core business process: Student Recruitment**

Use Cases (sub-processes) include:

- design student recruitment campaign;
- organise student recruitment event;
- issue recruitment materials;
- handle enquiries from prospective students;
- organise communications with students and potential students;
- analyse recruitment and retention data.

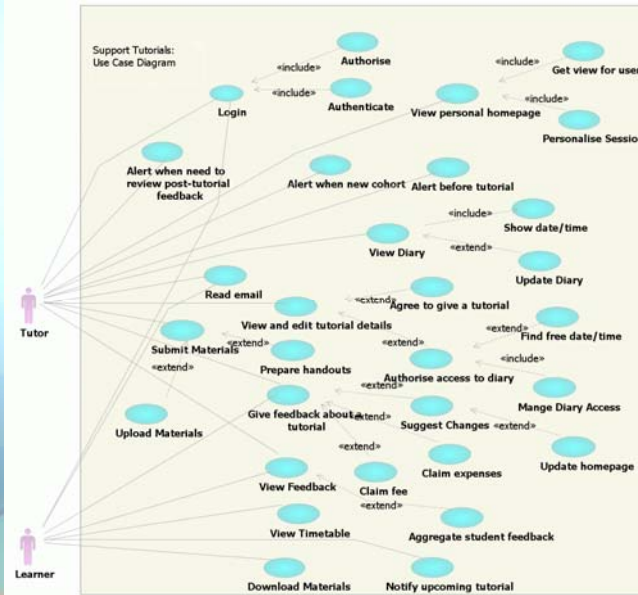
JISC Function & Activities Model

# Scope



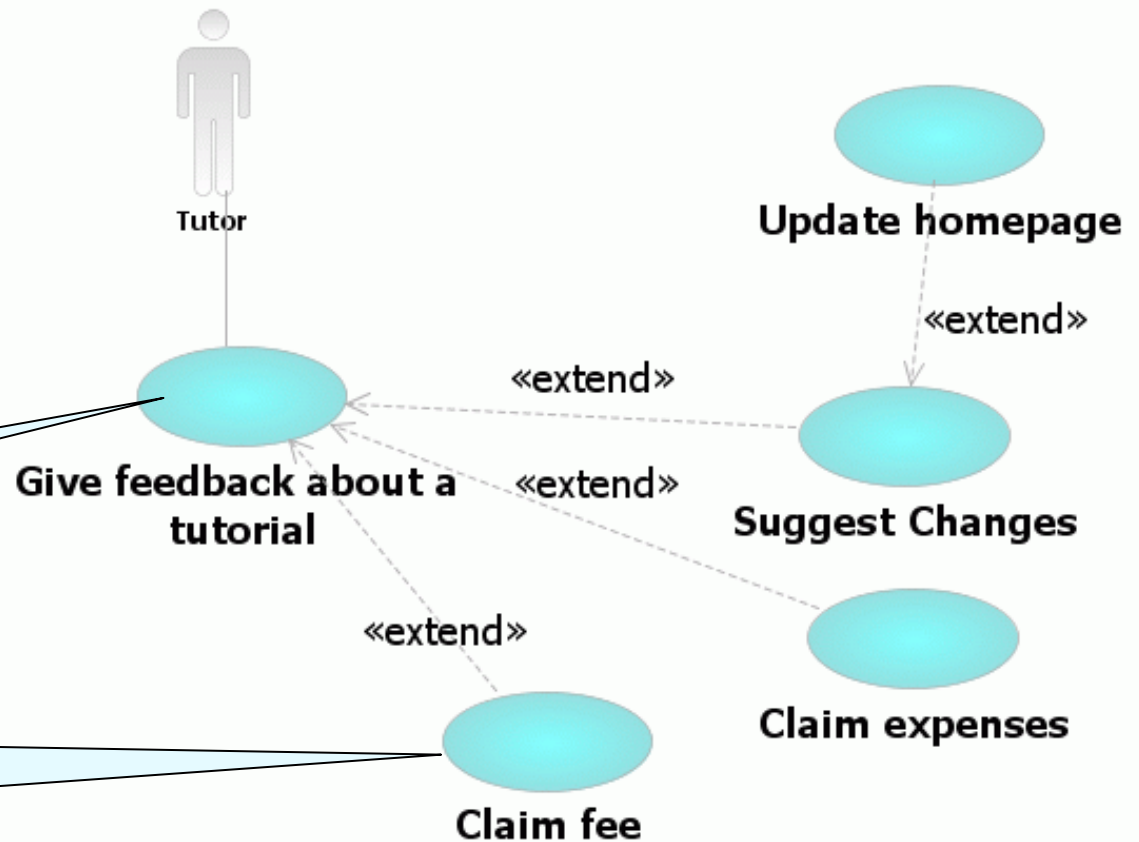
# Use Cases in a business process

A business process is a collection of 'units of functionality' or sub-processes or tasks, represented by Use Cases. Each Use Case may be detailed in a structured document and have its flow modelled by an Activity Diagram.



A Use Case from the business process

The Use Case sometimes has extensions



# Give feedback about a tutorial: a Use Case from the business process

## 1. Brief Description

Give feedback following a tutorial.

## 2. Basic Flow of Events

One day after the tutorial HORUS emails tutor to log on to home page and presents a form for immediate feedback on facilities and on learning behaviour of students, a form for claiming expenses and a form for claiming a fee. Another feedback form asks for any suggestions in changes to venue, timing, title, objectives and format of the tutorial and asks for commitment to give the tutorial again. Tutor agrees to give the tutorial at a future date and HORUS posts the responses to tutor's home page.

## 3. Alternative Flows

Tutor refuses to do future tutorials. Will be contacted for discussion.

## 4. Preconditions

Tutorial has taken place.

## 5. Postconditions

Feedback in teacher portfolio and available for QA

Agreement status for future tutorials known

Any fees or expenses claims in system

Tutorial details updated

## 6. Extension points

Claim Fees, Claim Expenses, Suggest Changes.

## 7. Special Requirements

Meet requirements of data protection act; of discrimination acts

## 8. Additional Information

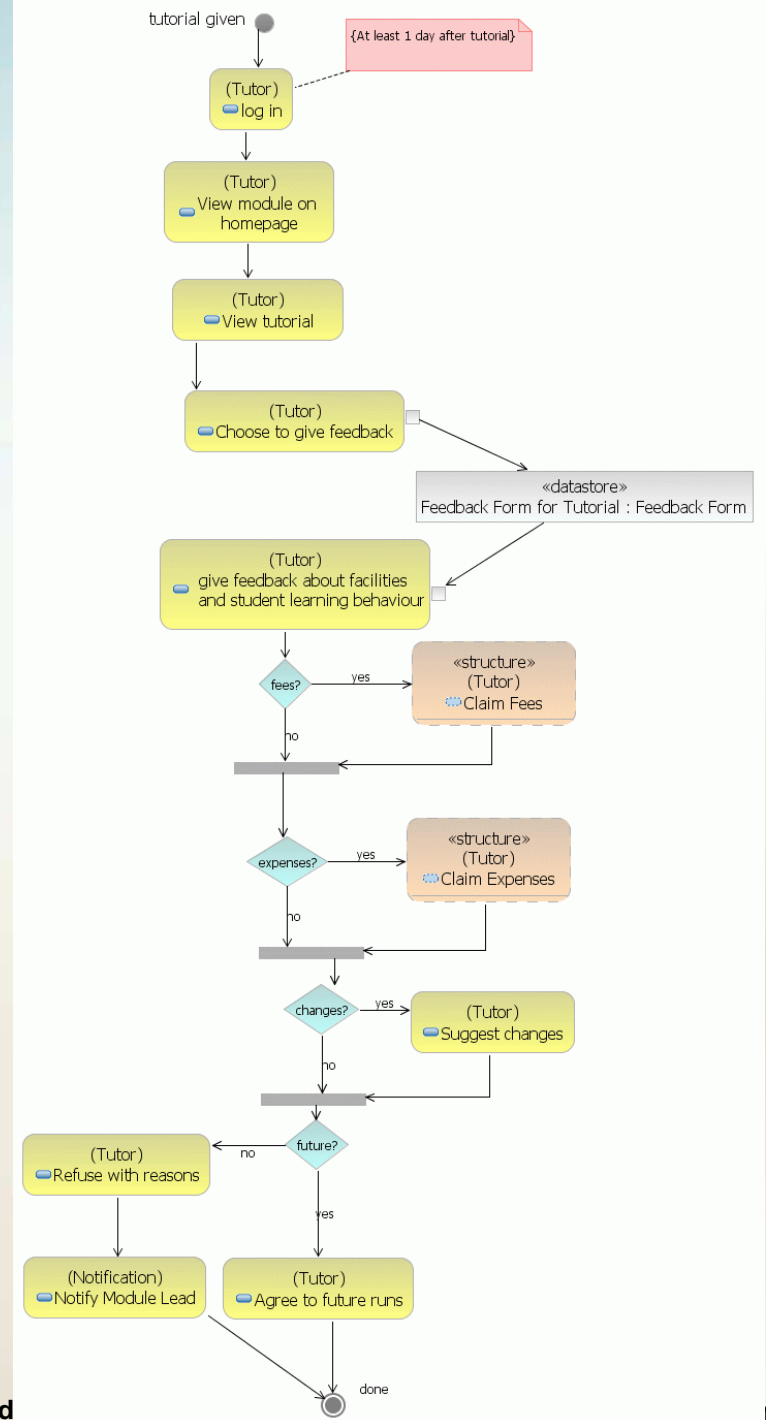
# Give feedback about a tutorial: Activity Diagram

Activity diagrams are typically used for business process modelling, for modelling the logic captured by a single use case or usage scenario, or for modelling the detailed logic of a business rule.

<http://www.agilemodeling.com/artifacts/activityDiagram.htm>

The primary consumers of Activity Diagrams are the customer stakeholder, testing team, and software development staff. For them, these diagrams form the visual "blueprints" for the functionality of the system, as described and detailed in the use cases. Tracing paths (threads of execution) through these Activity Diagrams enables all stakeholders in the process to understand and monitor the development of system functionality.

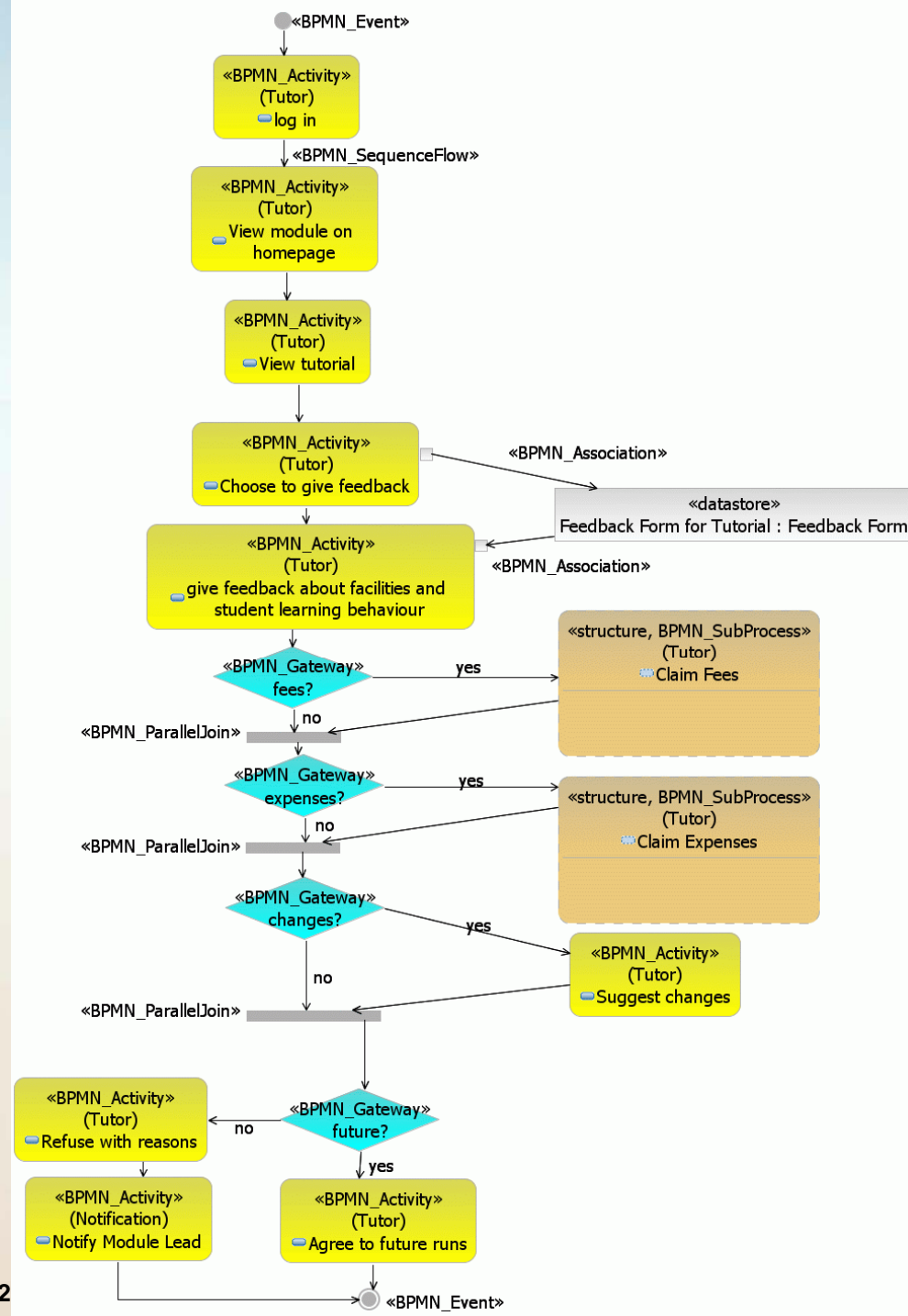
<http://www-28.ibm.com/developerworks/rational/library/content/RationalEdge/apr01/UMLActivityDiagramsApr01.pdf>



# BPMN

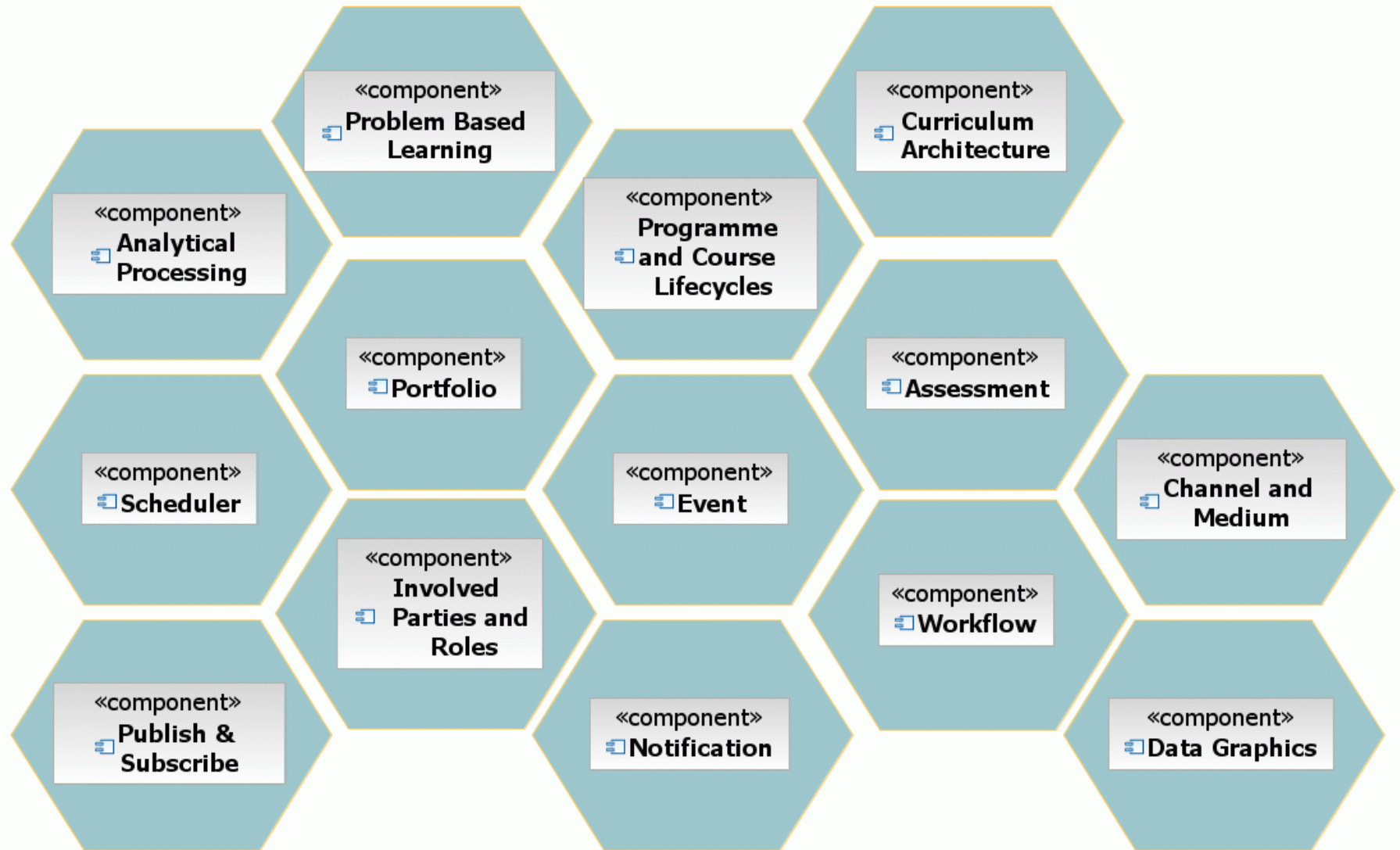
## The Business Process

**Modeling Notation (BPMN)** is a standardized graphical notation for drawing business processes in a workflow. BPMN was developed by Business Process Management Initiative (BPMI), and is now being maintained by the Object Management Group since the two organizations merged 2005.

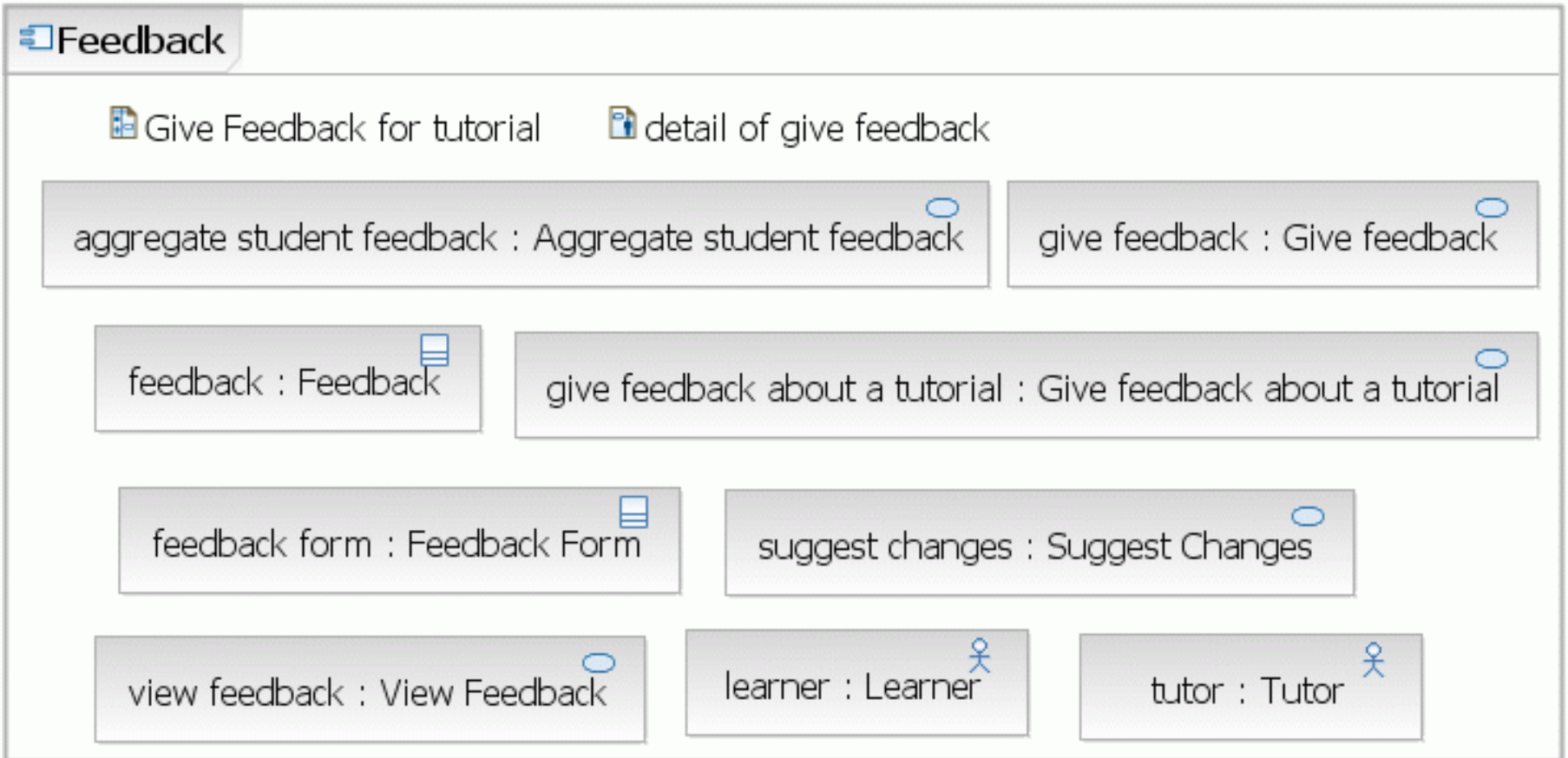




# Candidate components

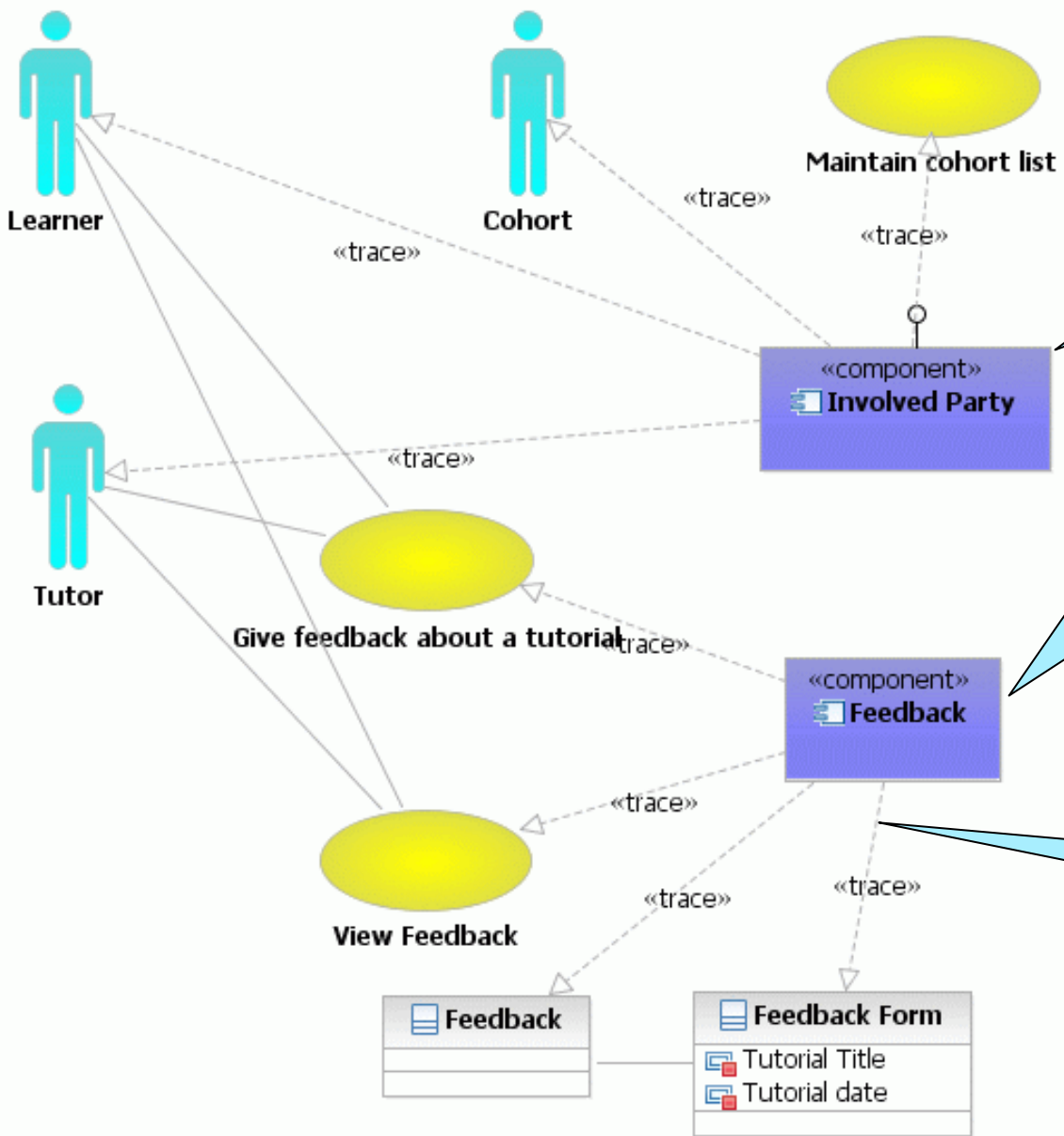


# Mapping Model Elements to Components



Application Architecture: Coverage (requirements traceability), interdependencies & reusability

# Requirements Traceability

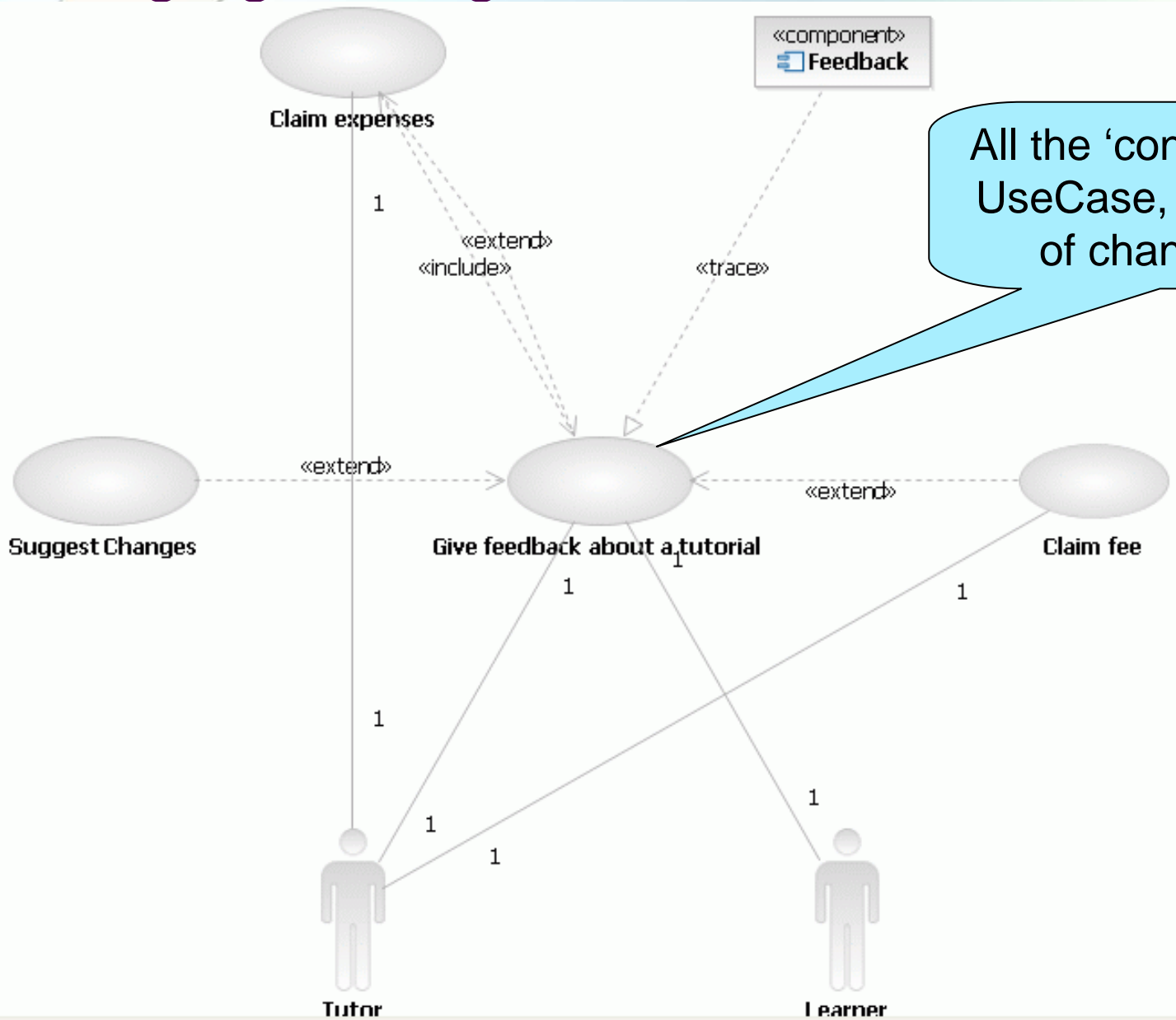


Component responsible for implementation of everything to do with **people and organisations**

Component responsible for implementation of everything to do with **feedback**

«trace» stereotyped relationship

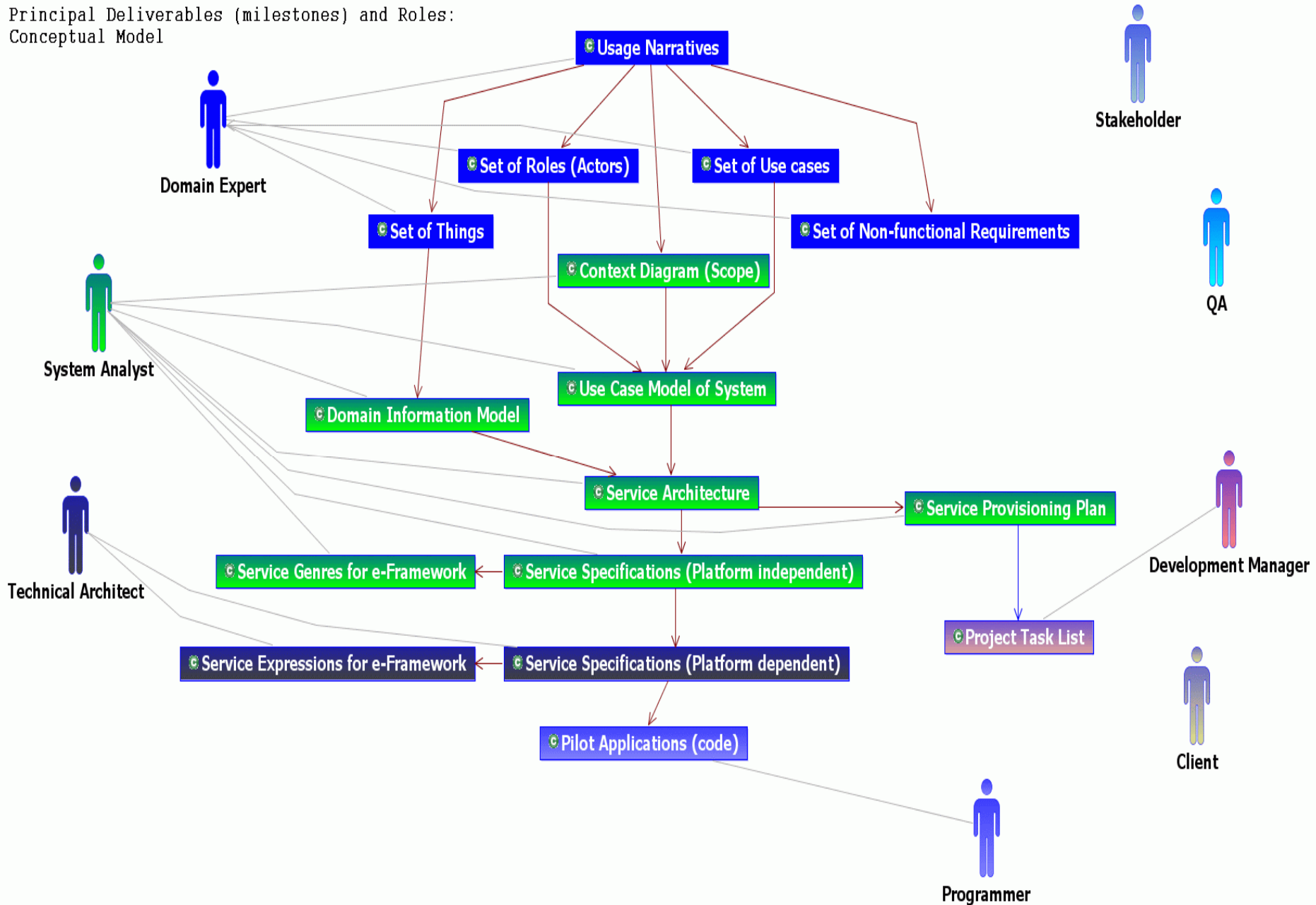
# Managing Change



All the 'connections' of a UseCase, hence a view of change impact

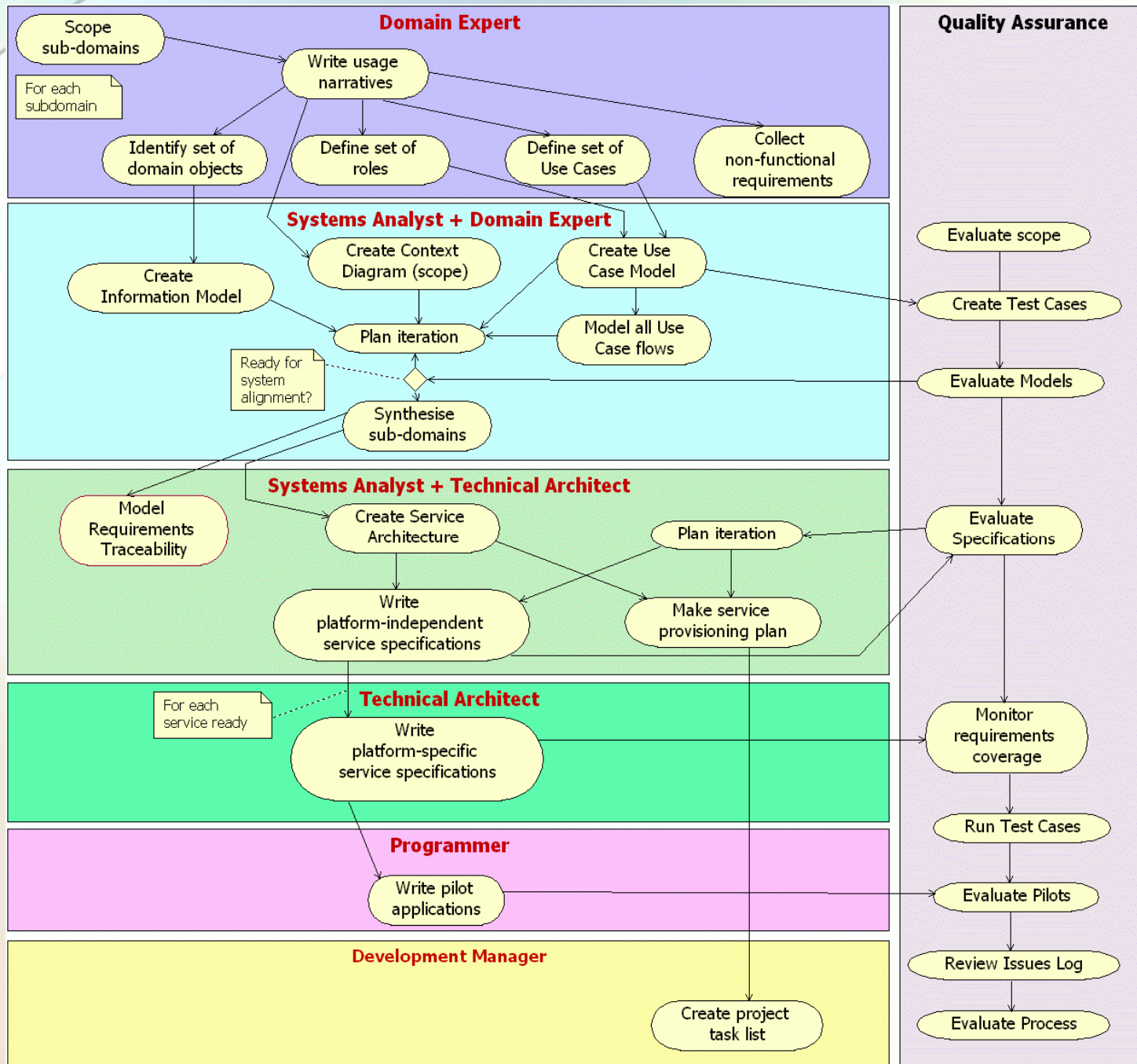
# The Development Process

Principal Deliverables (milestones) and Roles:  
Conceptual Model



# Workflow overview

Development is iterative, incremental and quality driven



# Supporting the developers

Providing a process driven knowledgebase (PDK)

**About a process:**

## About a step in a process:

- What do I do here?
  - Who else is involved?
  - What resources do I need?
  - What are the deliverables?
  - Are there tools for this?
  - Is there any guidance available (templates, best practice, examples etc.)?
  - What are the quality criteria for the deliverables?
- How does this process fit in with everything else that's going on?

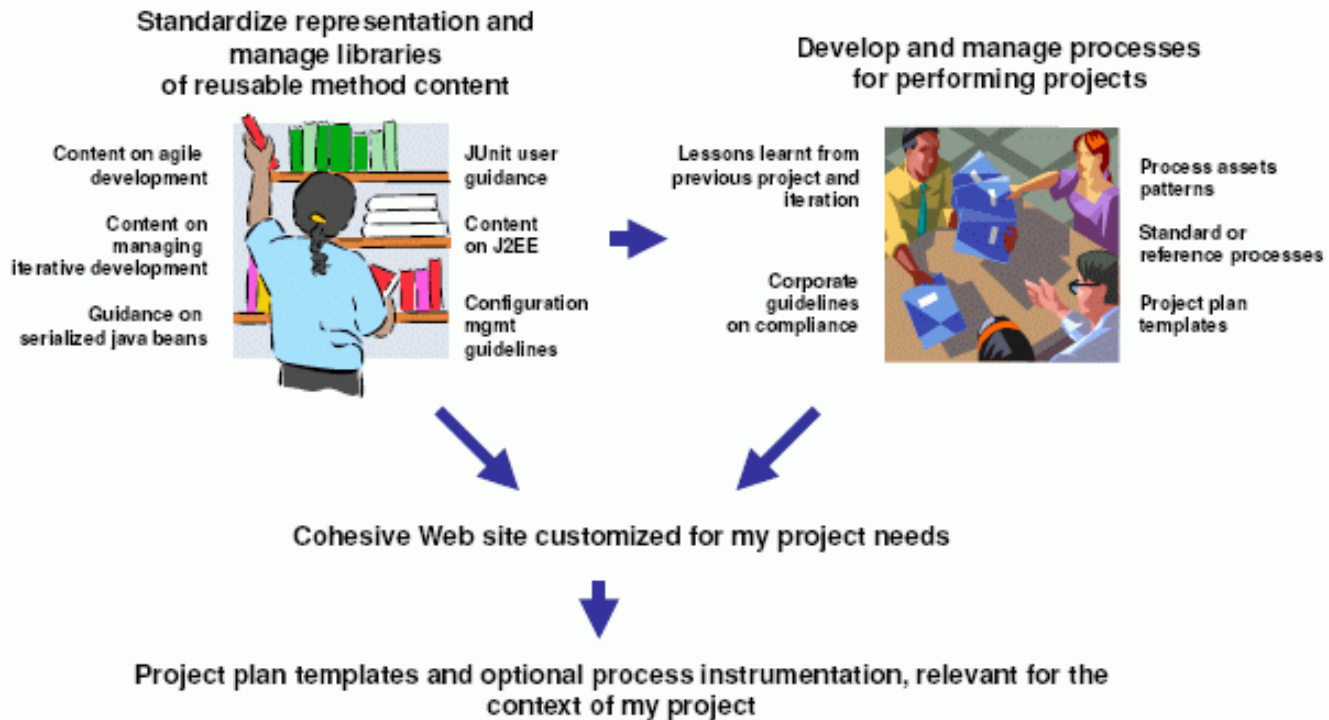
# Unified Modelling Architecture (UMA)

UMA is an architecture for the conceiving, specifying, and storing of **method** and **process** metadata.

(UMA is an evolution of the OMG industry standard Software Process Engineering Meta-model (SPEM))



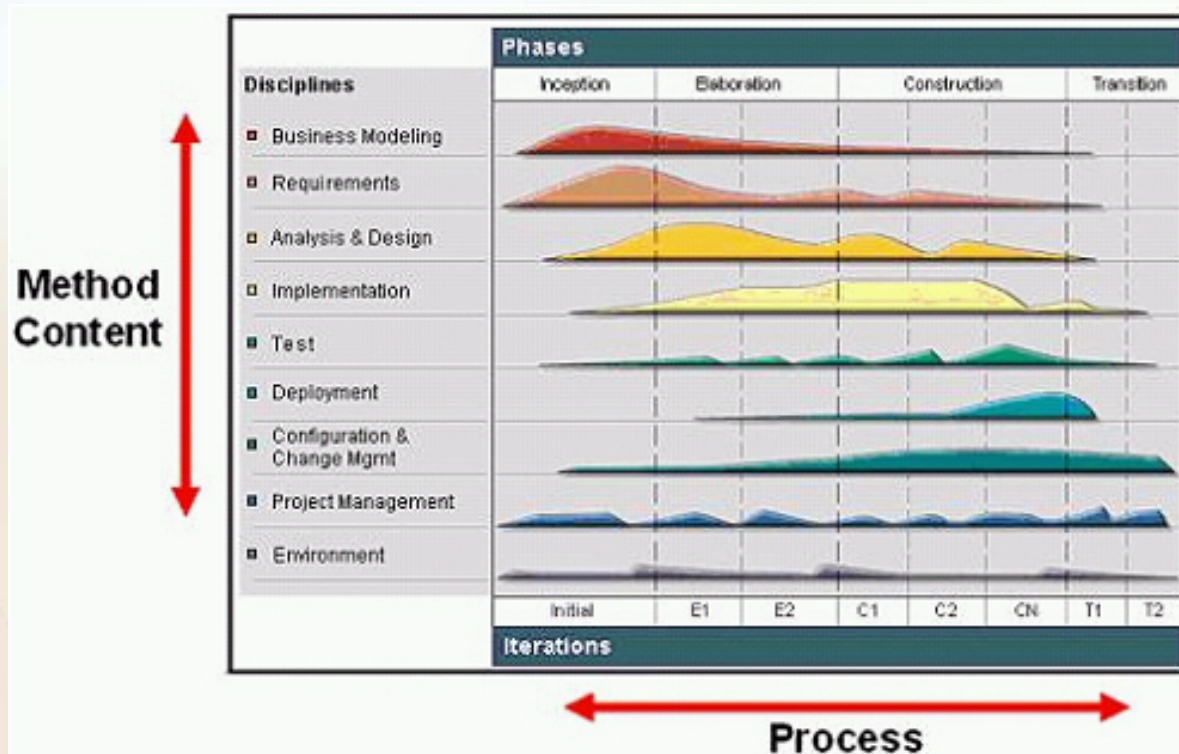
**How**



**When**

# Method Content vs Process

1. **Method content** describes what is to be produced, the necessary skills required, and the step-by-step explanation describing how specific development goals are achieved.
2. **Processes** take the method content elements and relate them into semi-ordered sequences that are customized to specific contexts.



# Guidance

(See handout)

Guidance is a general term for additional information related to [roles](#), [tasks](#), and [work products](#). Examples of guidance are:

**Guideline** - how to perform a particular task or grouping of tasks

**Template** - provides for a work product a predefined table of contents, sections, packages, and/or headings, a standardized format, as well as descriptions how the sections and packages are supposed to be used and completed.

**Checklist** - identifies a series of items that need to be completed or verified.

**Tool Mentor** - shows how to use a specific tool to accomplish some piece of work

**Supporting Material** - Used as a catch all for other types of guidance not specifically defined elsewhere

**Report** - A predefined template of a result that is generated on the basis of other work products as an output from some form of tool automation.

**Concept** - outlines key ideas associated with basic principles underlying the referenced item.

**Practice** - Represents a proven way or strategy of doing work to achieve a goal that has a positive impact on work product or process quality.

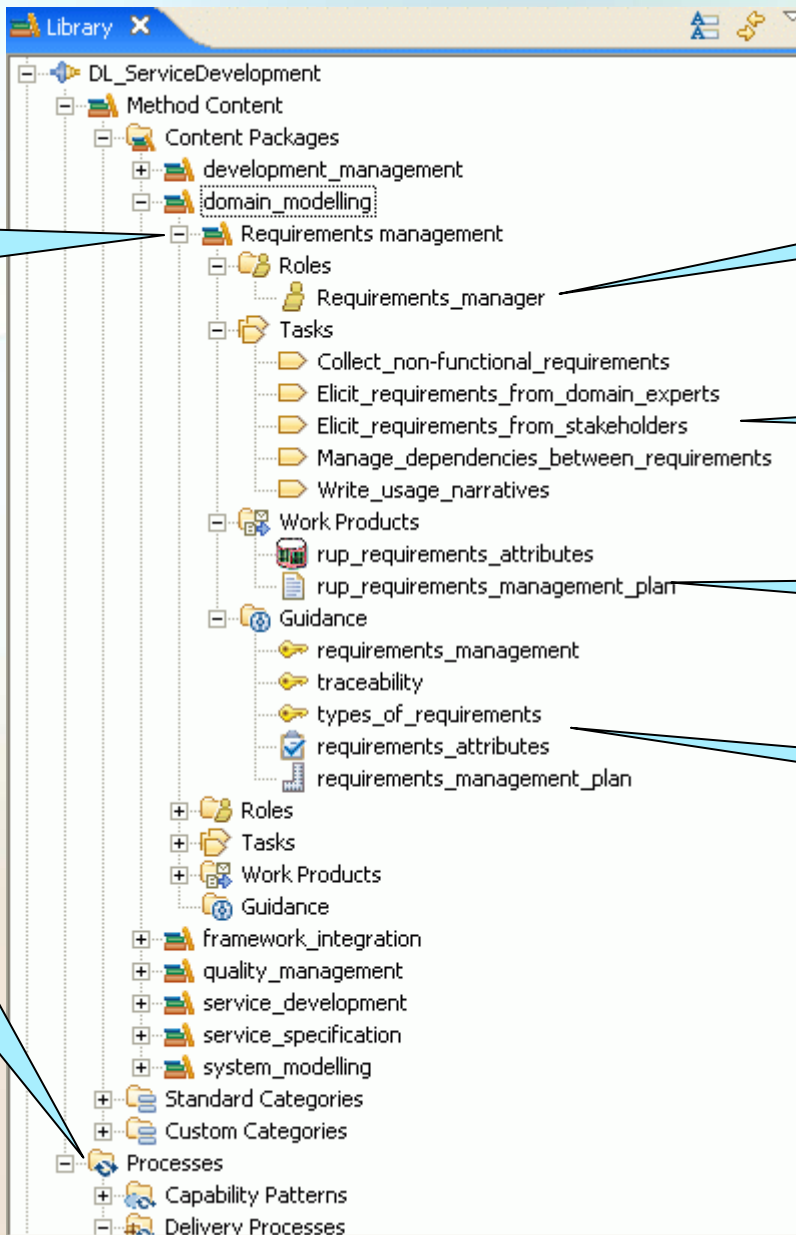
**Reusable Asset** - Provides a solution to a problem for a given context.

**Term Definition** - Defines concepts and is used to build up the Glossary.

**White Paper** - A special concept guidance that has been external reviewed or published and can be read and understood in isolation of other content elements and guidance.

**Example** - A specific type of guidance that provides an example of a completed work product.

# PDK Process Library



Method Content Package

Processes assembled as instances of method content

Roles

Tasks

Work Products

Guidance

# Using the PDK

## How do I model

Task: Manage

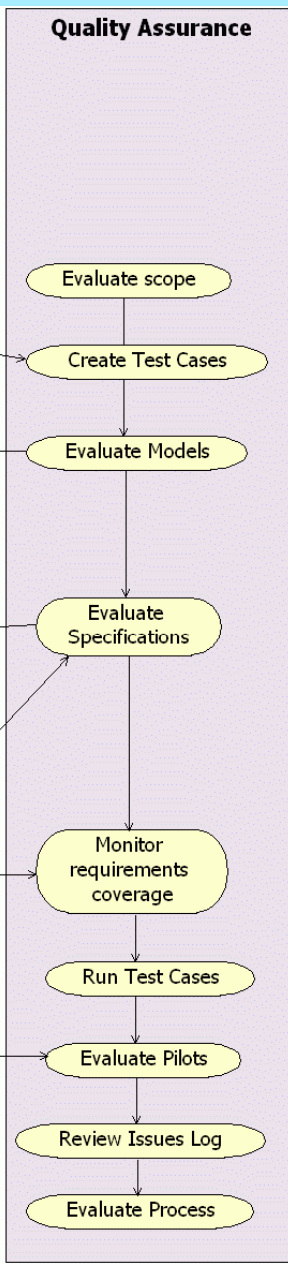
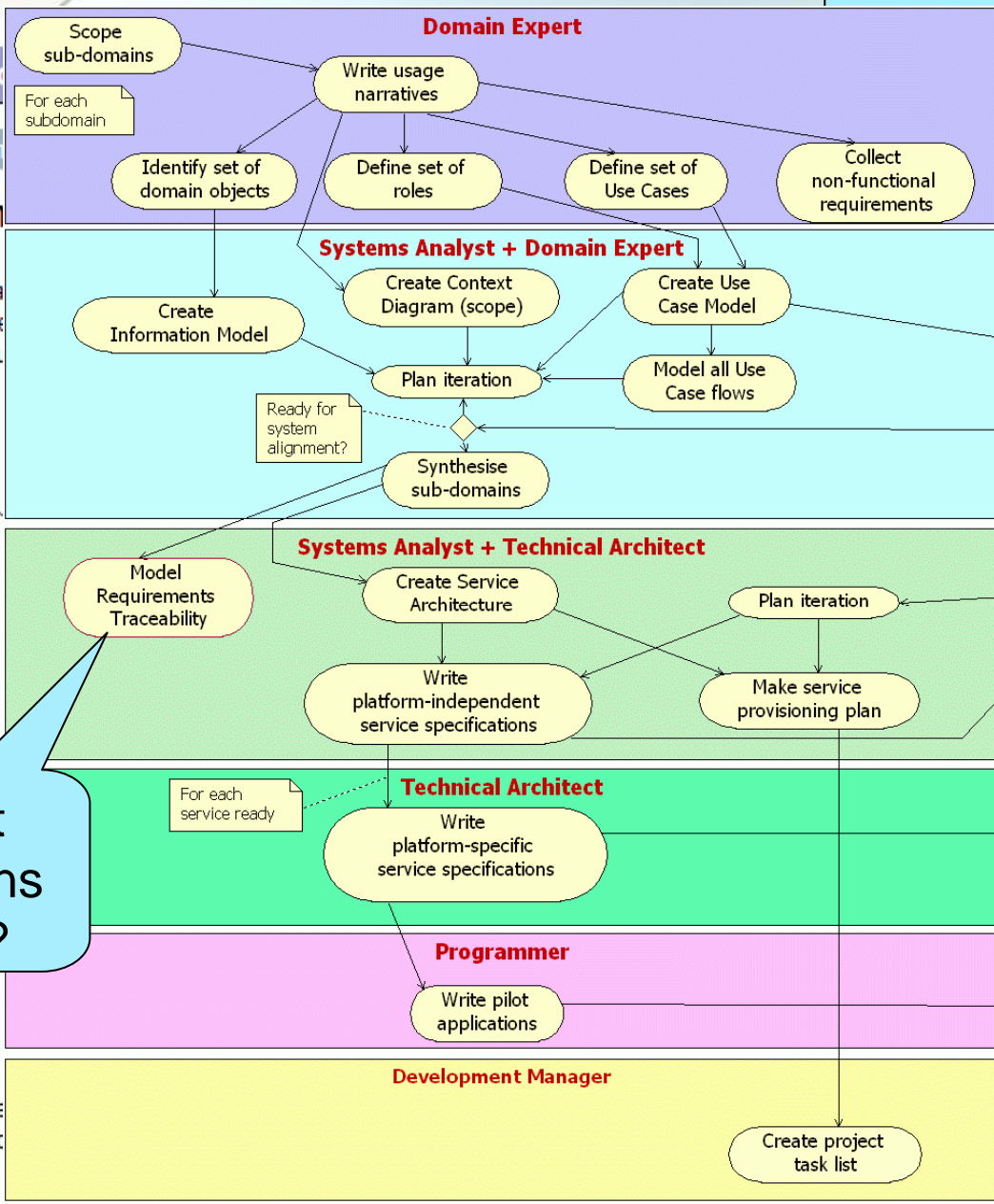
Main Description

### Introduction

Traceability is the ability to track the lineage of products, and end-user requirements.



What happens here?



ide?

elements involved in elements, test work

status, benefit, risk, etc.

Use Cases between

# Summary & Reflection

- Domain experts want and need to operate in their own domain with their own language;
- A defined, understood and agreed development process is helpful;
- Visual modelling in UML can be a communications enabler;
- Requirements must live throughout the whole development and product lifecycle;
- For real soa requirements must not be left behind but be an integral and traceable part of service specification and implementation
- Developers will be helped by having a process driven knowledgebase to hand.

# A few references

- The UML specification: .

<http://www.omg.org/technology/documents/formal/uml.htm>

- UML2 toolkit by [Hans-Erik Eriksson](#), [Magnus Penker](#), [Brian Lyons](#), [David Fado](#) , 2003, Wiley, ISBN: 0471463612

- Writing Effective Use Cases, Cockburn A. 1999,

<http://www.infor.uva.es/~mlaguna/is2/materiales/BookDraft1.pdf>

Thank you for listening

*Questions?*