

# Requirements Engineering: Stakeholders

CECS 590

# Organizational...

- Literature Reviews
  - Last minute questions
  - Last minute extension
- Team projects
  - Team members
  - System idea and added value
- Early feedback survey

# Recap time!

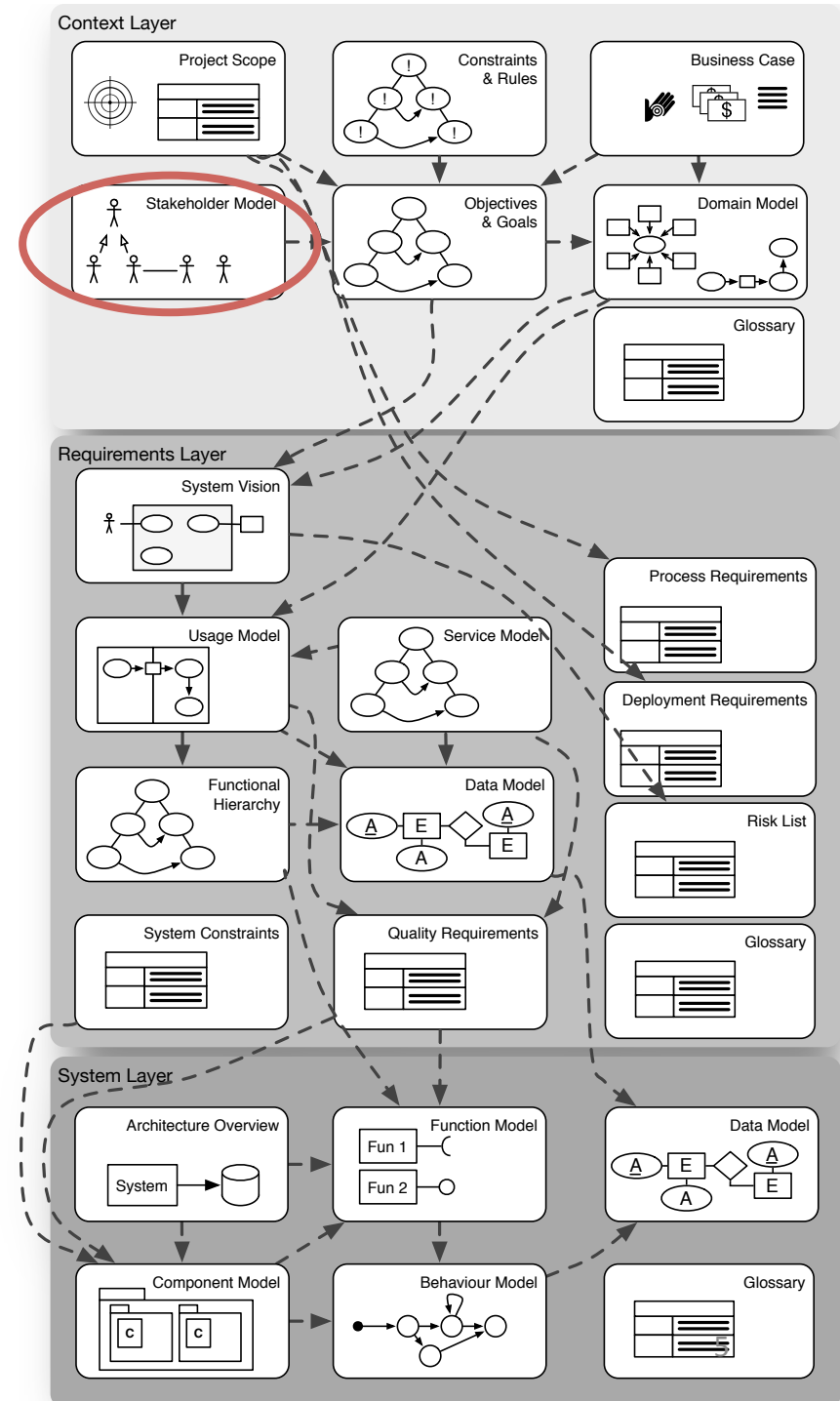
- Why business case analysis?
- What is a business case?
- How do I write one?
- What are the elements?
- What tools are available?

# Requirements Engineering – Outline

- WHY do we need Requirements Engineering and what is it?
- Principles: Definitions, process, roles, problem/solution view, artifact orientation
- System Models: Decomposition and abstraction, system views
- Frameworks: What reference structures can I use for requirements?
- Business Case Analysis: Why are we building this system?
- **Stakeholders: Who are the people to talk to about requirements?**
- Goals and Constraints: What are the major objectives for the system?
- System Vision: What exactly do we want to achieve?
- Domain Models: What are the surrounding systems ours interacts with?
- Usage Models: How will the system interact with the user?
- Software quality models: How to determine the quality characteristics?
- Quality requirements: How to specify which qualities need to be met?
- Process requirements: How to specify constraints for development?
- Towards a system specification: How to hand over to design?
- Quality assurance: How to ensure that RE is done in a good way?
- Change management: How to evolve requirements?

# Learning outcomes

- **Stakeholders**
  - Definition, characteristics, rationale
  - Relations with further Content Items
- **Elicitation Techniques**
  - Techniques for elicitation and negotiation
  - Typical problems in practice



# Stakeholders

- What is a stakeholder?
- Identifying stakeholders
- Classifying stakeholders
- Analysing stakeholders
- Managing stakeholders

# What is a Stakeholder?

## Definition

A stakeholder is a person or organization who influences a system's requirements or who is impacted by that system.

## Addition

Interest in the system does not necessarily mean interest in project success (Example: legislation, staff association)

# Critical points

- Role of Stakeholders with respect to the **system under development**
  - Relevant for project success
  - Priority of stakeholder
  - Type of interest
  - Skill in phrasing interests/requirements
  - ...
- Role of Stakeholders with respect to **project and project organisation**
  - Reporting
  - Decision competences
  - ...





# MULTI-CLASSING

Because wizards run out of spells

# Identifying Stakeholders

There are different possible approaches to identifying stakeholders, and most likely the best way to make sure all have been identified is a mix or iteration of these approaches, for example in the order they are presented in:

- **Phases:** Analyzing the aspects and development phases of software systems development to find the responsible roles. This approach is an easy way to set up early elicitation meetings with the most important, rather obvious stakeholders.
- **Reference list:** Instantiating generic reference lists of stakeholders (see below) for the concrete project context. This second step takes standard roles into account that have been included in reference models and enhance the initial quick list of stakeholders. One simple reference model that is being used in software engineering to map out stakeholders is the so-called *Onion Model* with its four concentric spheres: product, system, containing system, and the wider environment.
- **Context:** Inspecting the business and operational context of the system under development, and understanding which concrete roles are involved. This step makes sure that the specifics of the project under consideration are all met and special roles are considered.
- **Goals:** Iteratively analyzing and refining a generic goal model and deducing the related roles. This approach is especially suitable for finding passive stakeholders that do not have an active interest in issuing own goals, but whose constraints have to be adhered to, for example legislative representatives.

# Classifying Stakeholders: Roles

There are different types of and roles for **stakeholders** with **different interests** and, consequently, **differing requirements**:

- Owners (e.g., individual, shareholders, even the public)
- Partners (e.g., other companies providing related services)
- Department heads, managers
- Staff, developers
- Regulatory bodies, legislative representatives
- Suppliers
- Customers
- Competitors

# Classifying Stakeholders: Functions

All of these stakeholders hold one or more functions with regard to the software system under development:

- Decision makers (e.g., sponsors, artefact approvers)
- Information providers (e.g., domain experts)
- Regulatory (e.g., legal body)
- Implementers (e.g., developers, testers)
- End users
- Post-implementation support (e.g., trainers, managers)

# Analysing Stakeholders

Stakeholder analysis includes developing a stakeholder model and a stakeholder matrix that gives an overview of the following information:

- What is their **motivation**?
  - How much power or influence do they have?
  - Are they supporters or detractors of the project?
  - Are they fully engaged, or disinterested?
- What are their **expectations**? (needs, concerns, wants)
- What do we expect of them? (task and result)
- How **expert** are they at what they do?
- Where are they? What is their **availability**? (for communication and/or negotiation)
- What is their **authority**? (level of influence, impact, or enhancement)
- What is their **relation** to other stakeholders? (hierarchical and official relations as well as informal friendships or rivalries)

This list is a simple version of a **stakeholder analysis template**. Using such a template ensures that the major characteristics and needs of a stakeholder are taken into account.

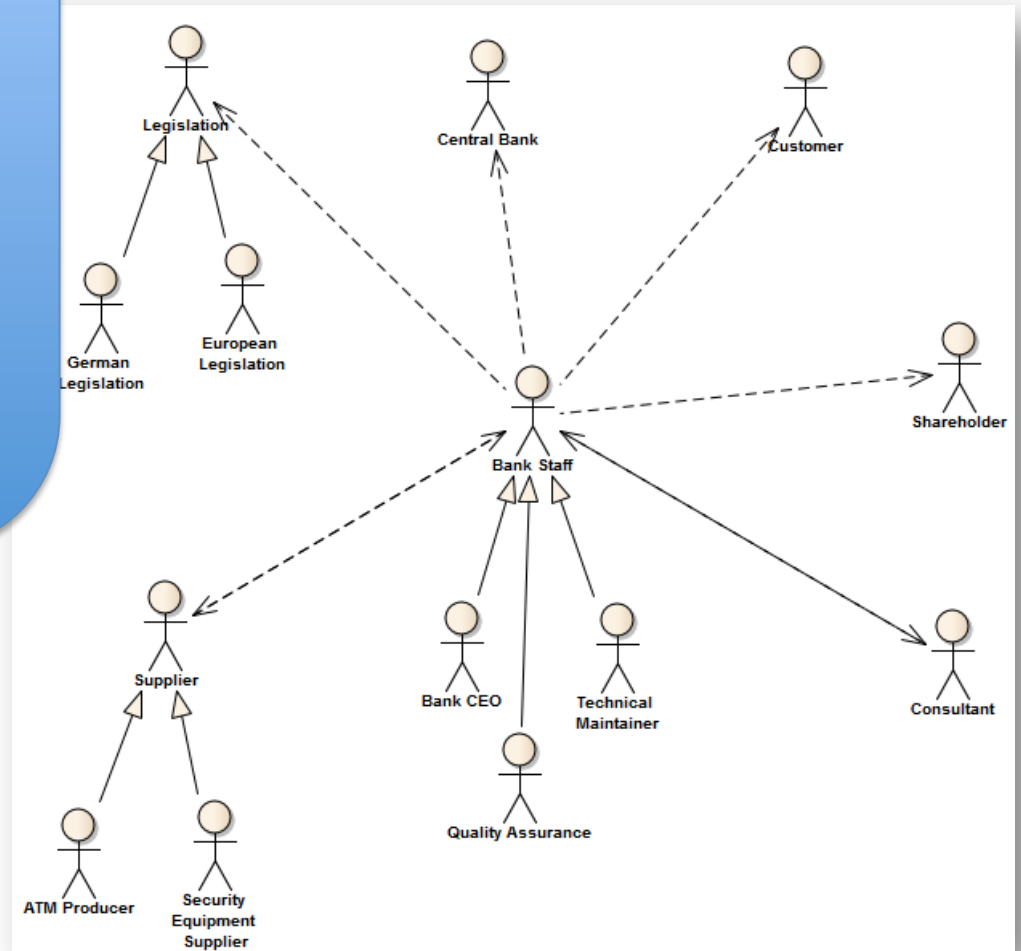
# Stakeholder Model

Fix arrows!

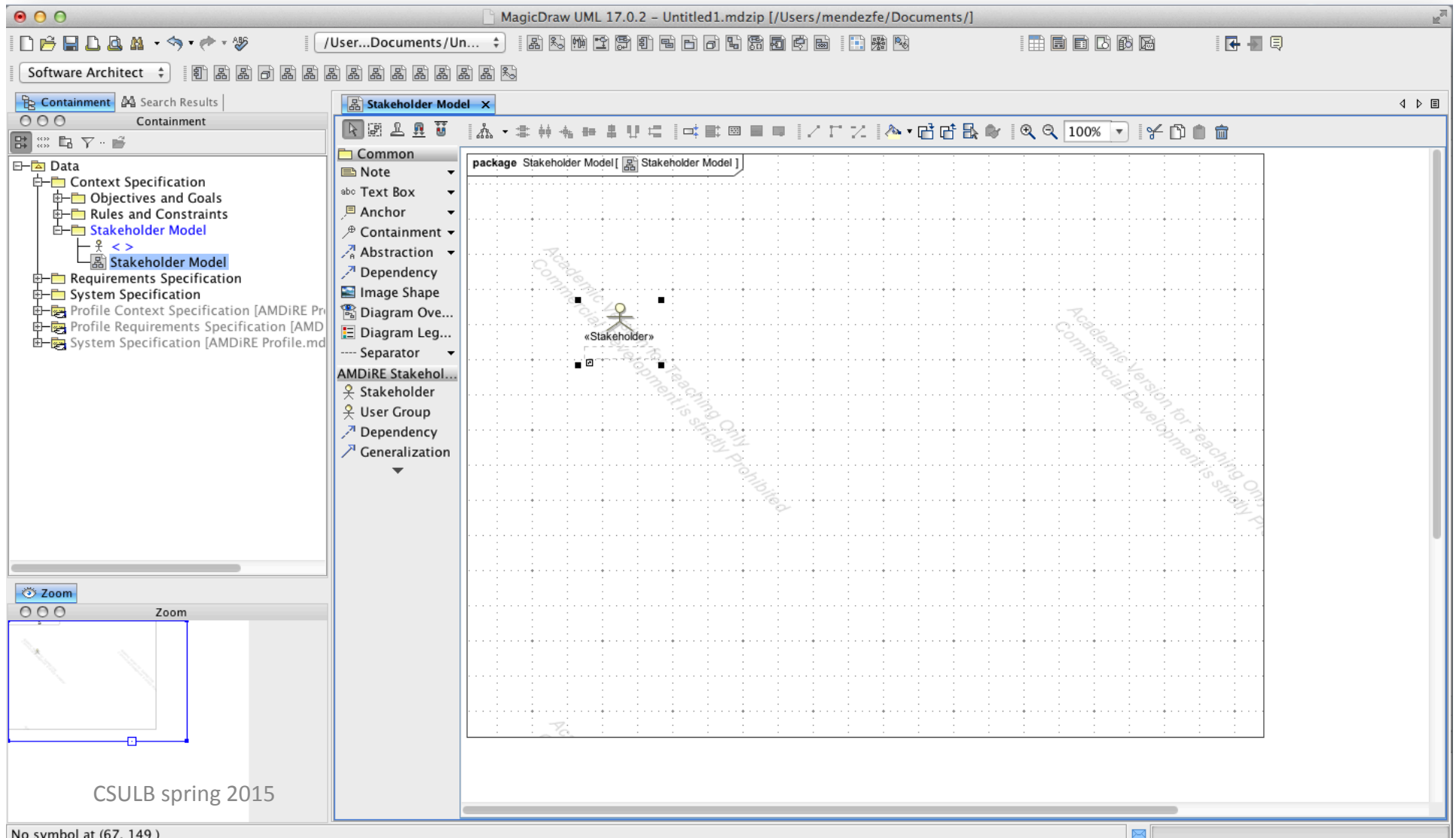
Make a legend and label them  
Arrows should have different meanings,  
and all of them should be labeled.

User groups are a specialisation of stakeholders interacting with the system.

Means to document Stakeholder Models are UML Actor Hierarchies, informal hierarchical graphics or natural text.



# Elaborating a Stakeholder Model



# Elaborating a Stakeholder Matrix

Alexander & Robertson  
“Understanding project sociology  
by modeling stakeholders”

CSULB spring 2015

Stakeholder role (the job title, department, or organization that indicates a stakeholding)	Stakeholder name (the name(s) of the responsible stakeholder(s))	Necessary involvement (estimate of when and how much time)	Classes of Knowledge						
			Business Goals	Business constraints	Technical constraints	Functionality	Look and feel	Usability	Performance
Client									
Customer(s)									
Business/ subject experts									
Future idea specialists									
Current system specialists									
Clerical user									
Technical user									
Potential user									
Sales specialist									
Marketing specialist									
Aesthetics specialist									
Graphics specialist									
Usability specialist									
Safety specialist									
Security specialist									
Cultural specialists									
Legal specialists									
Environmental specialists									
Maintenance specialists									
Packaging designer									
Manufacturer									
Product installer									

**Figure 3. A fragment of a stakeholder analysis template. The complete template contains many more stakeholder roles and knowledge classes.**



# In-class exercise

- Let's find 25 stakeholders for a car sharing system.

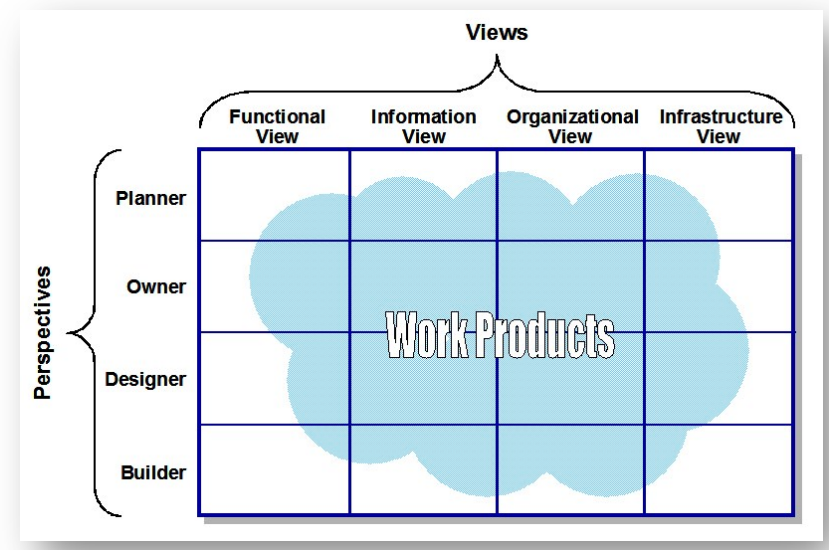


# Managing Stakeholders

- The most important factor in stakeholder management is communication.
- The challenge is to establish a stakeholder management strategy that addresses the stakeholders' information needs: who, what, how, when and why.
- In order not to overwhelm the stakeholder with information, concise communication is preferable, and for successful communication it is crucial to encourage feedback.
- There are a number of different communication means that might be suitable, for example consider workshops, interviews, phone calls, web, etc. A blend of what works best depends on the project settings but it has to be ensured that communication is two-way.
- Once communication is well established, the information flow needs to be kept up to date in the stakeholder model or matrix – whichever representation was chosen and works best.

# Viewpoints

- The stakeholders' different viewpoints support communication and requirements elicitation.
- Viewpoints are related to view-based frameworks, for example architecture frameworks.
- A number of perspectives onto the system leads to a more complete overall understanding.
- At the same time, the discussions help to detect inconsistencies and contradictions.
- This is done via arranging the stakeholders into groups that serve for systematic requirements elicitation.
- The development of the requirements in models enables to document differences and contradictions and provides a common ground for solving them.
- Finally, the models are consolidated in a homogeneous view. This is also known as View Point Resolution.

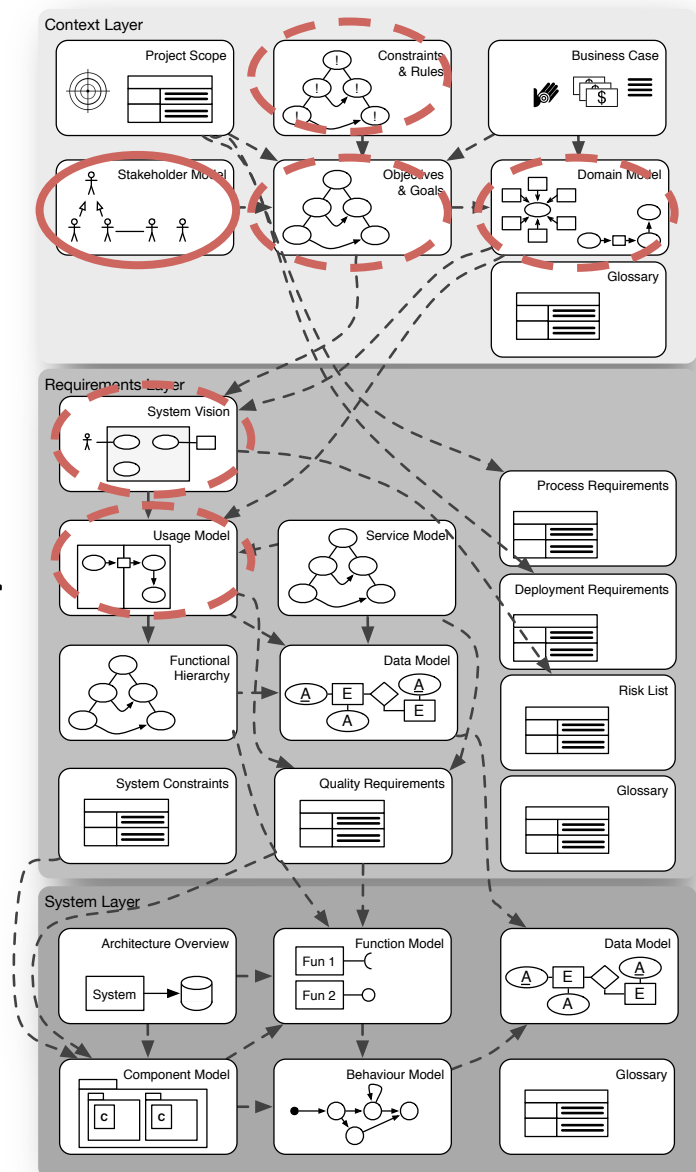


# Requirements elicitation techniques

- As is-analysis of existing systems
- Analysis of business processes
- Analysis of usage scenarios
- Deriving requirements from goals
- Interview / workshop with stakeholders
- Observation of users during work, site visit
- Questionnaires
- Analysis of documents
- Templates
- Prototypes
- Experts
- Creativity techniques

# Relations between Stakeholders and other Content Items

- Stakeholders are an essential part of Requirements Engineering
- They pursue goals, include users of the system, impose constraints, ...
- A Stakeholder Model is the basis for
  - Rationale, Requirements source
  - Goal Modeling
  - Domain Model
  - System Vision
  - Usage Model (Use Cases / Scenarios)



# Stakeholder revisited

- Examples
- Template
- Creativity Techniques for Requirements Elicitation

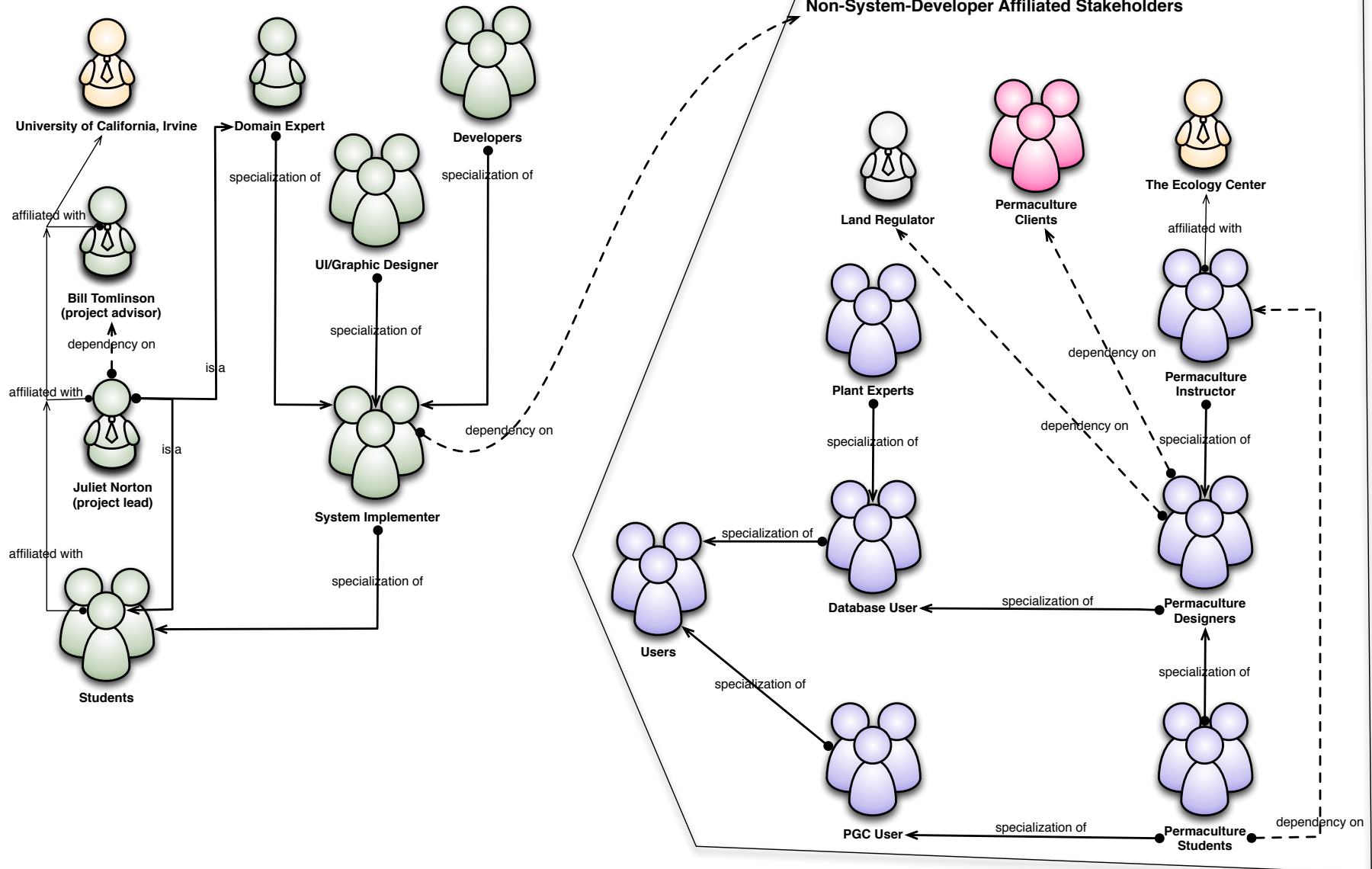
# Examples

- From my graduate course at UCI last year

# Examples

## The Plant Guild Composer: Stakeholder Model

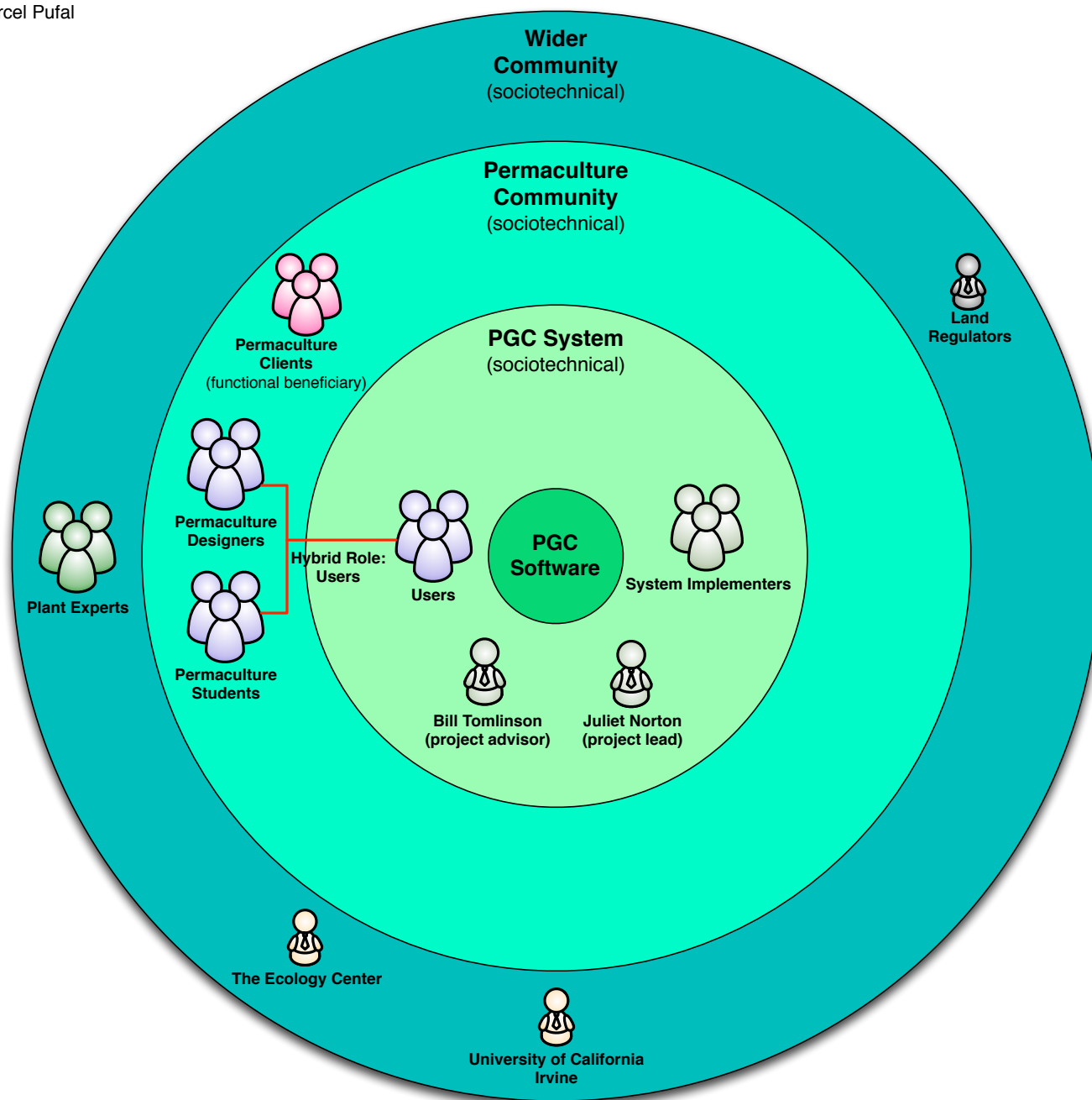
Juliet Norton and Marcel Pufal





## The Plant Guild Composer: Stakeholder Model

Juliet Norton and Marcel Pufal





# Requirements Specification: Story of Stuff – Citizen Muscle Bootcamp

# Examples

## EXECUTIVE SUMMARY

The Story of Stuff – Citizen Muscle Bootcamp (hereby referred to as “the system”) is a Massive Open Online Course that will allow users to participate in learning about the different causes that threaten the sustainability of our world.

Currently, the world is faced with multiple threats that harms its sustainability, on both an environmental level as well as a societal level. People play a part in this due to their lack of understanding and knowledge of how their actions affect those around them. The system is an attempt to change this situation by educating as many people as possible about this issue, by creating an online platform that allows everyone to take courses to learn about the various effects of their decisions.

We will begin with an introduction that describes the current problem and how the system will attempt to solve it, as well as some basic discussions of the resources, risks, and assumptions of this project. We then follow this with an analysis of the stakeholders that may be affected either directly or indirectly, through an onion model and a stakeholder functionality table. Following that, we dive in to a discussion of the various goals of the system, which are classified as either a business, usage, or system level goal. From there, we combine aspects of these models into one high level picture in the system vision, which depicts the relationships and interactions of all stakeholders and their goals and concerns. The usage model is then used to detail out the interactions between a stakeholder and the system, through a use case diagram, a use case description, and a message sequence chart, which is a detailed diagram that then depicts the flow of processes in a particular use case. Finally, we conclude with a description of non-functional requirements, which are requirements that apply at an overall system level. These typically include process requirements, deployment requirements, system constraints, and quality constraints.

The ultimate goal of the system is to generate awareness of the issue, educate and inspire change amongst people who previously had no idea about the issue, and/or pool people together, encouraging collaboration to start making a change to the way we do things.

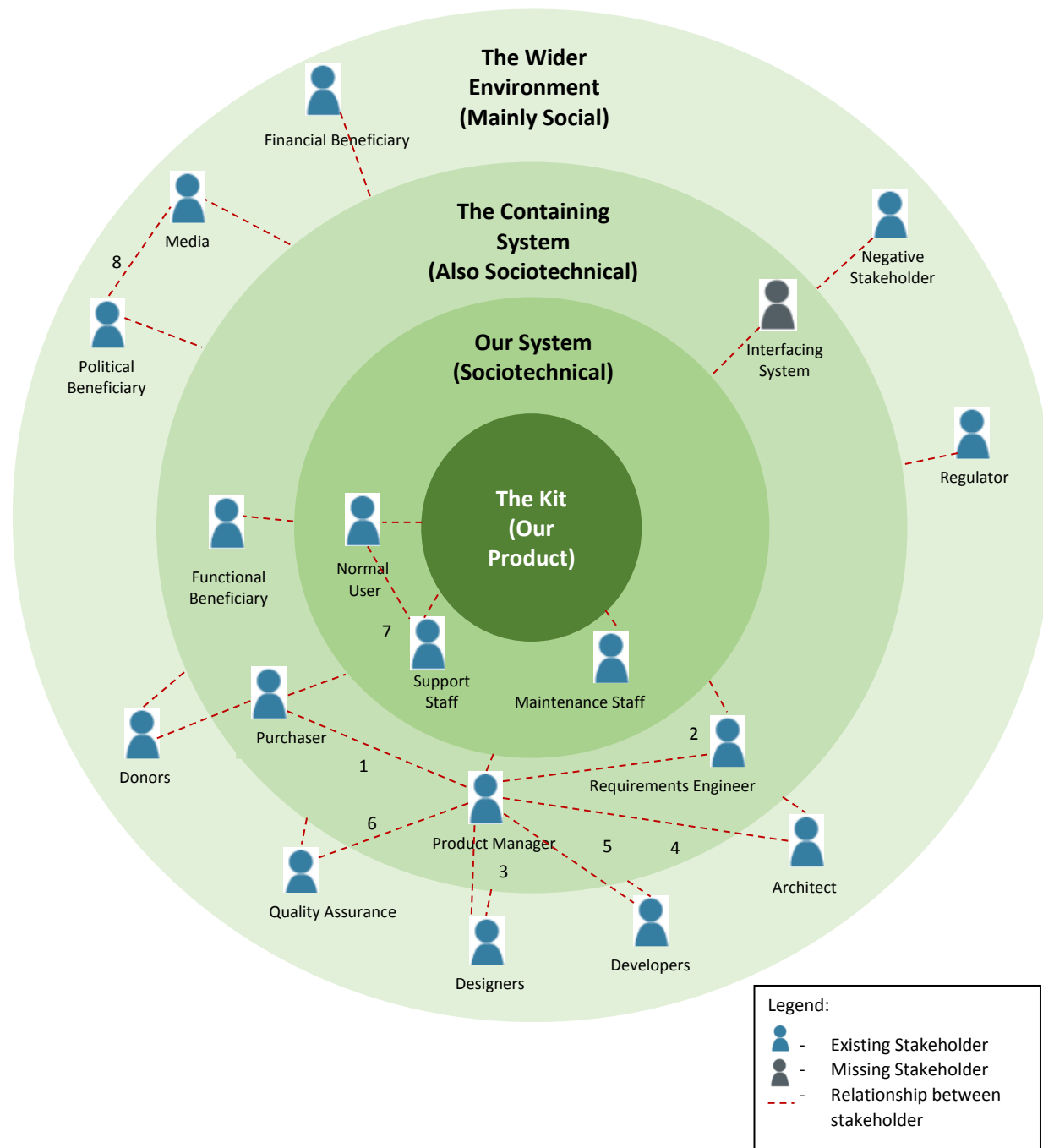
# Examples

## STAKEHOLDER MODEL

A stakeholder is any entity in the environment of the system who has an effect on the requirements of the system and who is impacted either directly or indirectly by the success or the failure of the system. The stakeholders in the system are represented and analysed using the onion model which has the following features:

1. This model has a complete focus on the system “Story of Stuff – Citizen Muscle Bootcamp” by placing it in the center of the diagram, which makes it more efficient than any of the stakeholder models available as this diagram makes the requirement engineer keep the system in mind at all times.
2. Our system contains spaces for all the stakeholders who maintain the system using the fixed processes and the ultimate users of the system.
3. The containing system contains all of those aspects which are outside the control of the system, but they do ultimately have an effect on the system, such as the impact due to the non-operational beneficiaries.
4. The wider environment is the section which is used to represent all those stakeholders who have an impact on the decisions made on the system, general legal and regulatory.
5. The normal operators are the users who perform various functions to use and operate the system.
6. Supporting roles are those beneficiaries which are placed on the bottom of the diagram and they carry a large portion of the weight of the system.
7. The head and shoulder icons are used to represent the stakeholders and the distance of the stakeholder from the core of the diagram showcases the impact that the stakeholders have on the system, the closer the stakeholder, the more impact they have on the overall system.
8. The roles for which the stakeholders have been identified are displayed in blue and for the roles where the stakeholders are absent are displayed in grey.
9. All the stakeholders have a dotted line indicating that they have a stake in the ground.

Figure 1: Onion Model



# Examples

## Stakeholder Definitions

The following is the description of the stakeholders for the “Story of Stuff – Citizen Muscle Bootcamp” which have been indicated in the onion model above.

### Normal Operator

The system contains normal operators which are any users of the system. They can be further classified into those users who either interact directly with the system or those users who have an actor to interact with the system on their behalf. The users and actors are further classified into active and passive users, passive users being those users who go the system and learn the materials provided in the system, but do not actively bring about any change and active users being those users who learn the material taught by the system and go ahead to actively bring about a change in the world.

### Support Staff

The support staff are those stakeholders who the users can contact in case they have any difficulty in using the system and require further assistance in using the system.

### Maintenance Staff

The maintenance staff are those stakeholders who are responsible to keep the system updated with new material, new technology and new statistics so that the system does not become obsolete.

### Functional Beneficiary

The functional beneficiaries are those entities who are impacted indirectly by the success of the system. These entities can be either the factory workers who would be paid more and would get a better working condition due

# Examples

Table 1: Stakeholder Functionality

Stakeholder Role	Functionality
New User	<ol style="list-style-type: none"><li>1. Register</li><li>2. See course list</li><li>3. Contact support</li><li>4. Read about current issues</li></ol>
Registered User	<ol style="list-style-type: none"><li>1. Login</li><li>2. See course list</li><li>3. Contact support</li><li>4. Read about current issues</li><li>5. Select course domain</li><li>6. Enroll for course</li><li>7. View course material</li><li>8. Upload podcast</li><li>9. Take survey</li><li>10. Connect people with similar interests</li></ol>
Support Staff	<ol style="list-style-type: none"><li>1. See course list</li><li>2. Answer questions</li><li>3. Provide feedback on questions</li></ol>
Maintenance Staff	<ol style="list-style-type: none"><li>1. Update current issues</li><li>2. Update course material</li><li>3. Review and upload selected podcasts</li></ol>

# Examples

## Stakeholder Relationships

The following are the relationships that are highlighted in the onion model:

1. There exists a relationship between the purchaser and the product manager as the purchaser would first go to the product manager to ask for the system to be built and the product manager would get the final built system to the purchaser.
2. The relationship between the product manager and the requirements engineer indicate that the product manager would work very closely to with the requirement engineer to gather the requirement from the stakeholders keeping the business goals in mind.
3. The relationship between the product manager and the designer indicate that the product manager would give the requirements to the designer and get the system designed which will be in turn returned to the product manager for the further stages.
4. The product manager would then give this design to the architect who would model various infrastructural alternatives keeping the constraints in mind and then would select the best alternative as indicated by the link between the product manager and the architect.



# In-class exercise

- Team up with your neighbor
- Develop a stakeholder model for an online auctioning system (like Ebay)

