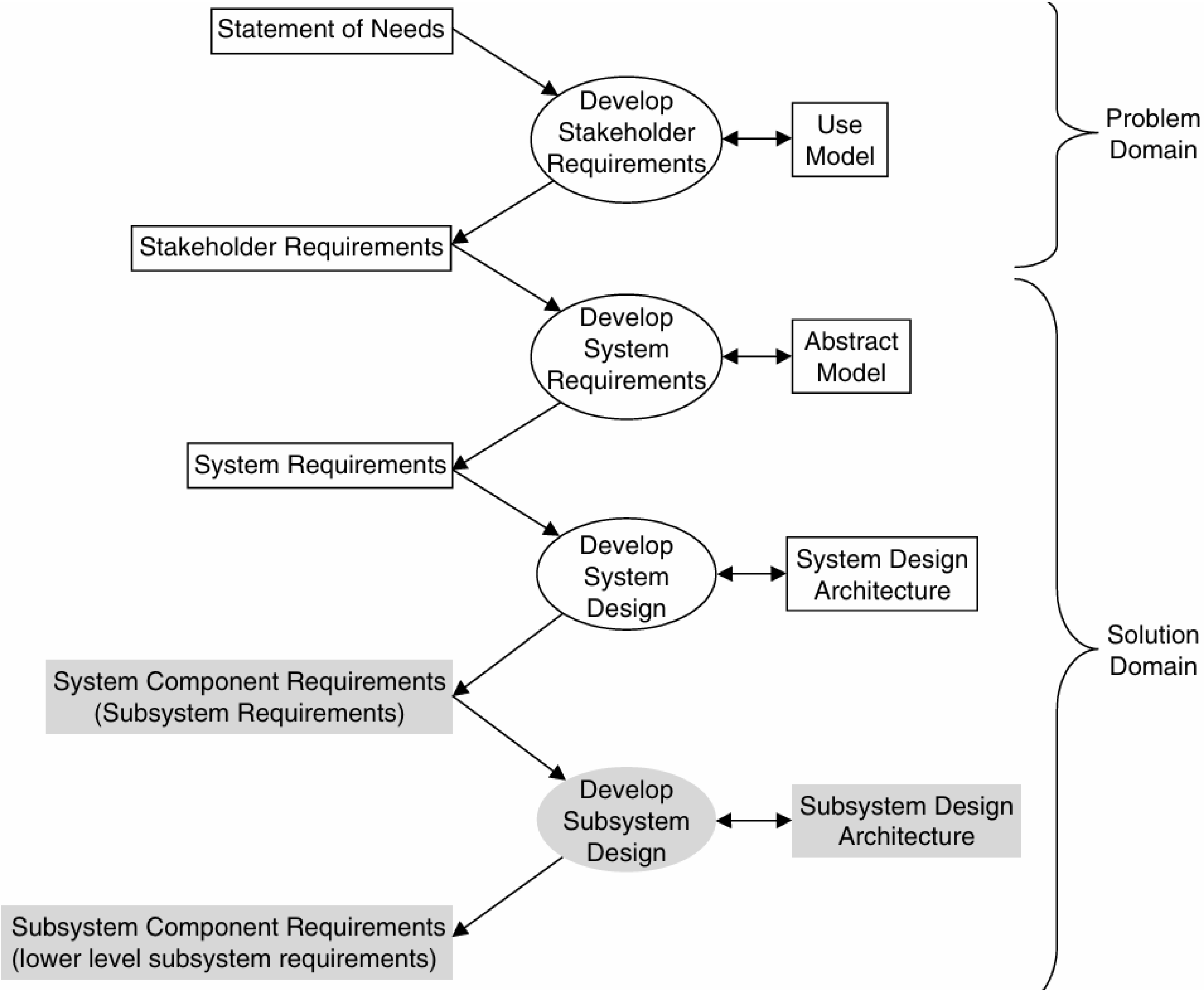
LECTURE 4

Software requirements Engineering

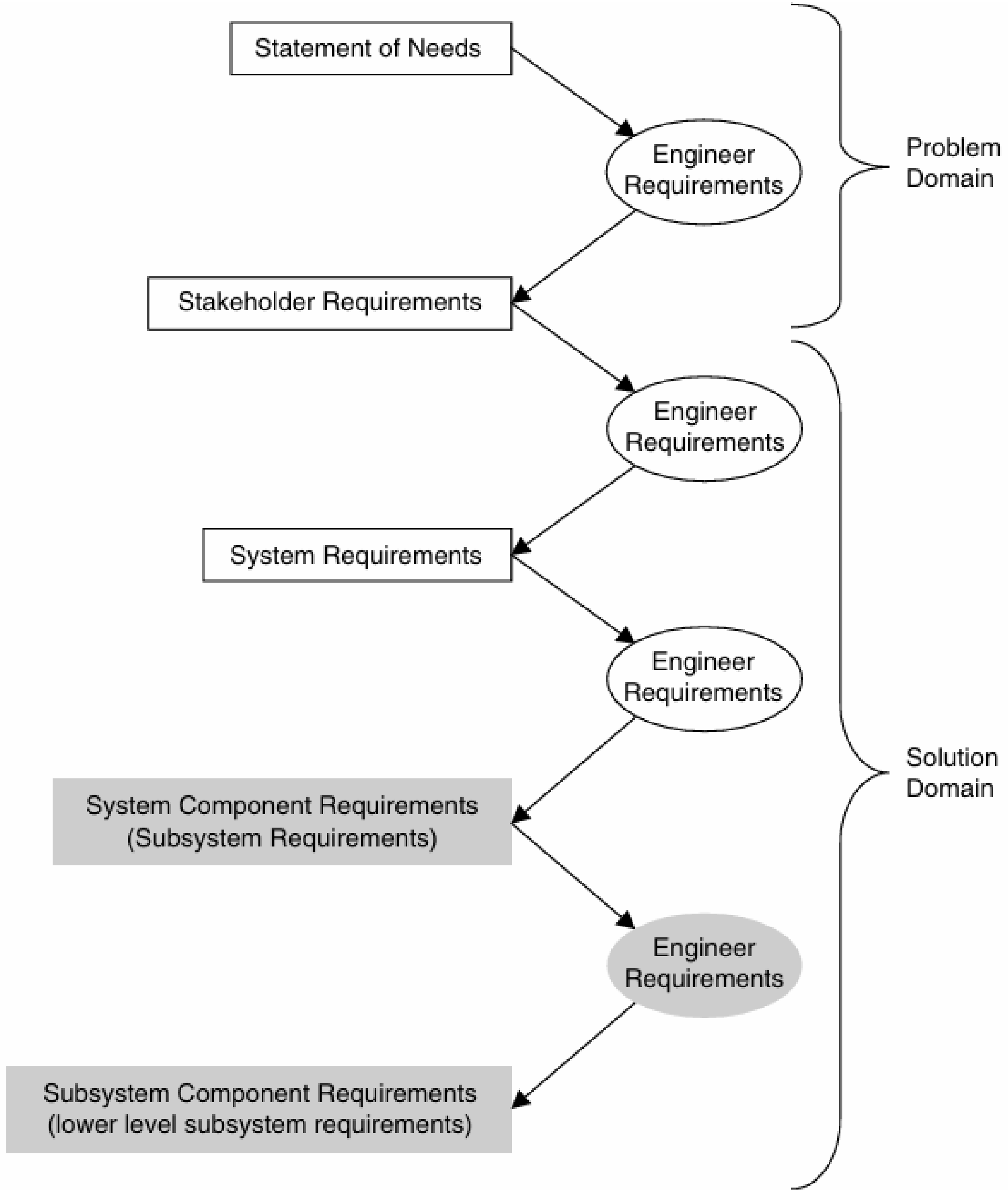
Generic Process Context

Instructor:
Dr. Natalia Chaudhry,
Assistant Professor, PUCIT, University of the Punjab, Lahore.


Statement of Needs
Develop Stakeholder Requirements
Use Model
Stakeholder Requirements
Develop System Requirements
Abstract Model
System Requirements
Develop System Design
System Design Architecture
System Component Requirements (Subsystem Requirements)
Develop Subsystem Design
Subsystem Design Architecture
Subsystem Component Requirements (lower level subsystem requirements)
Problem Domain
Solution Domain

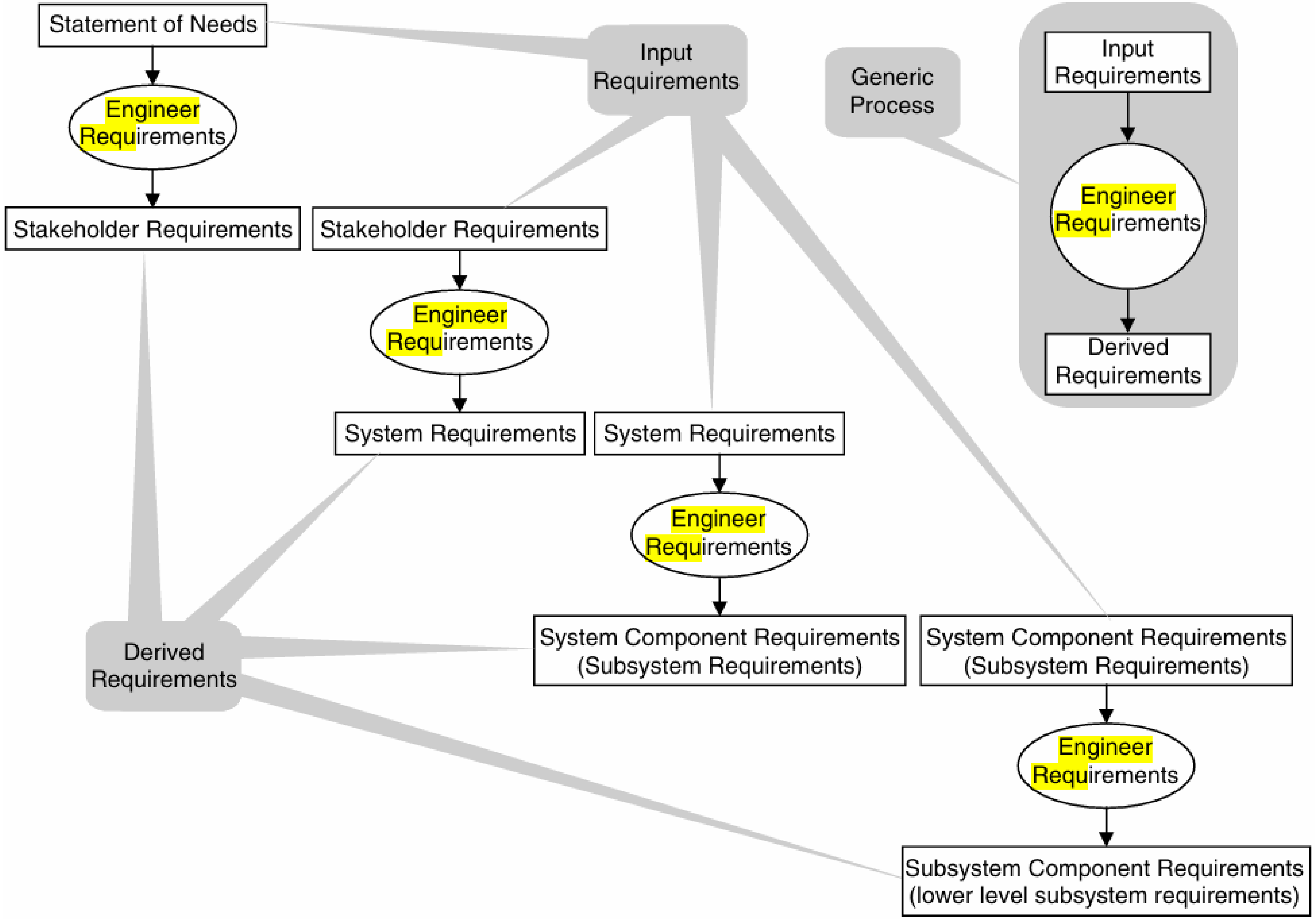
alternative way of considering the development process

Different levels of requirements engineering



alternative way of considering the development process, another view

Identifying input and derived requirements of the generic process



Requirements derived by one process become the input requirements of another process and leads naturally to the idea that the generic engineer requirements process takes in input requirements and generates derived requirements

Acceptance Criteria and Qualification Strategy

Before moving on to explain the internal details of the engineer requirements process, it is necessary to consider another class of information that is both an input to the process and derived by the process. This is information concerning the **qualification strategy for the requirements**

Testing is just one type of qualification strategy. Others include trials, certification and inspections.

The type of qualification strategy to be used will depend on the nature of the system; for example, systems that have safety critical aspects will have to be checked much more carefully than, say, a management information system.

Generic Process Information Model

Before considering the subprocesses within the generic engineer requirements process, it is useful to introduce a generic information model that supports the process.

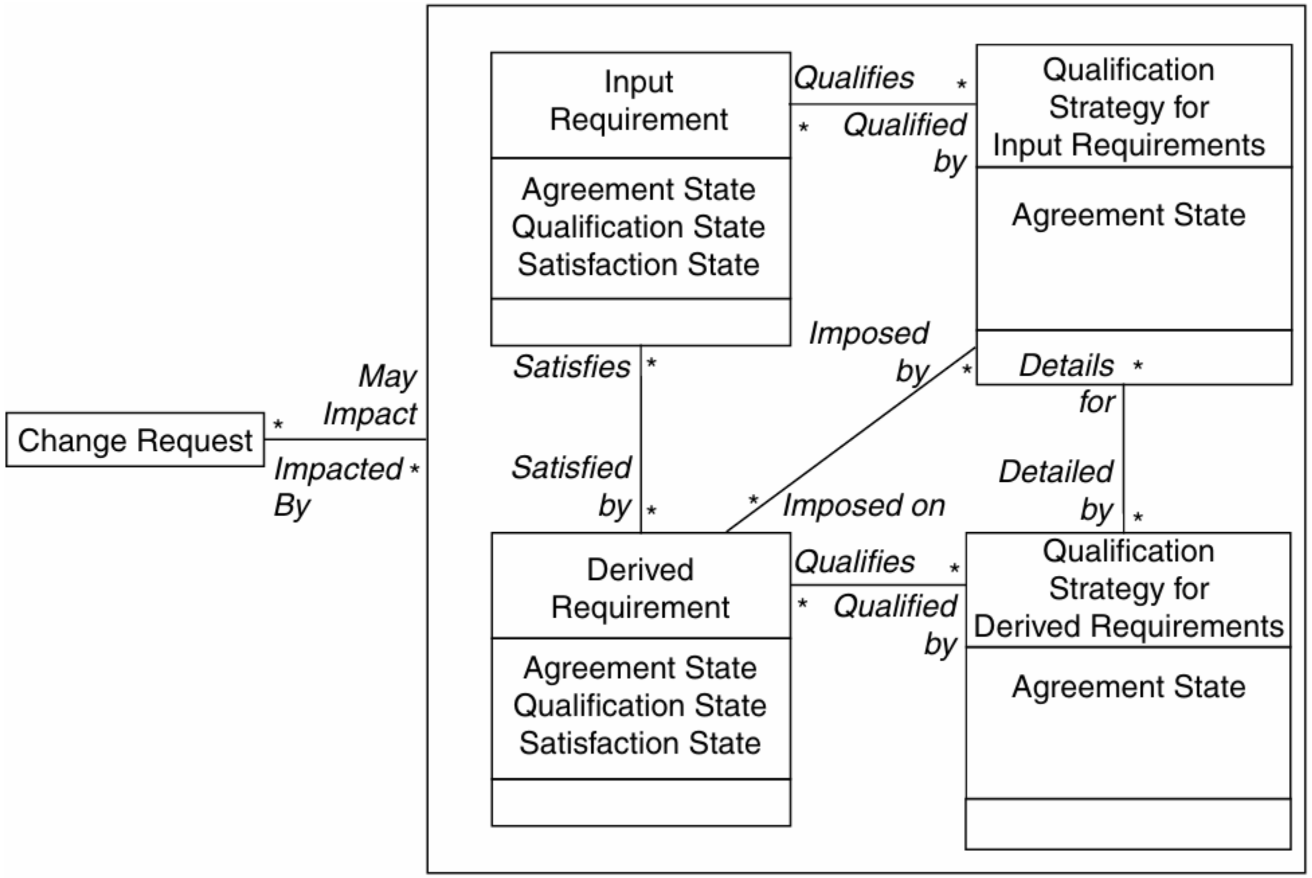
- The diagrams used to represent the generic process contain both process symbols and data or information symbols. The diagrams indicate, via the arrows, which information is being generated and used by each process.
- The purpose of an information model is to indicate what types of information exist and whether relationships can or should exist between the items of information.
- It is also useful to introduce state transition diagrams to indicate how the state of each type of information can be changed as time proceeds.
- Consequently, these state transition diagrams can give a visual indication of when and how processes interact with each other via the information.

Information Classes

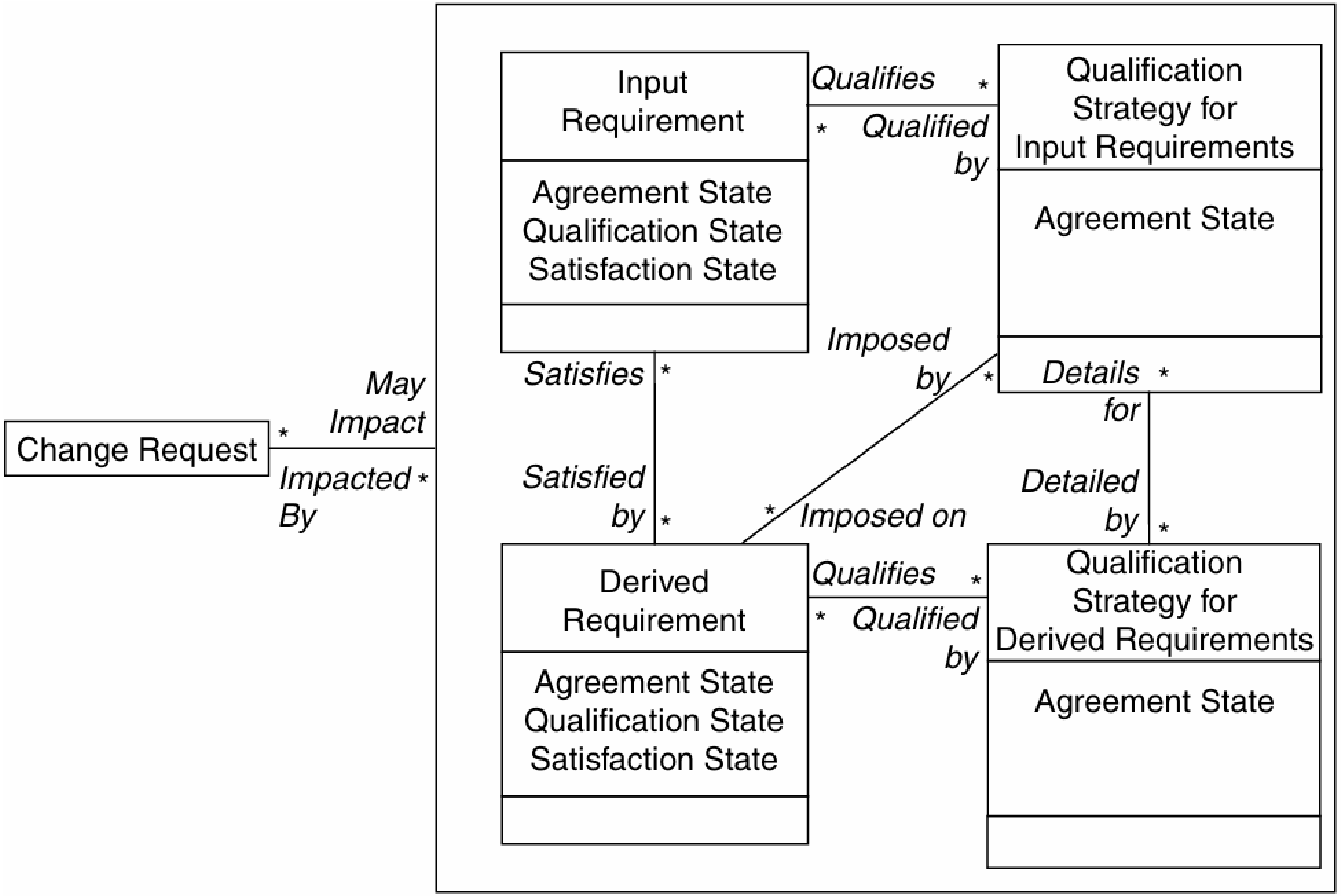
Information types already encountered in the generic process context include:

- input requirement
- derived requirement
- qualification strategy for input requirements
- qualification strategy for derived requirements
- change request.

The name of the class is always shown in the uppermost section (or only section) of the class symbol. The middle section (if present) indicates the names of attributes that the class can have. The bottom section (if present) contains any operations (often called “methods”) that can operate on the class.



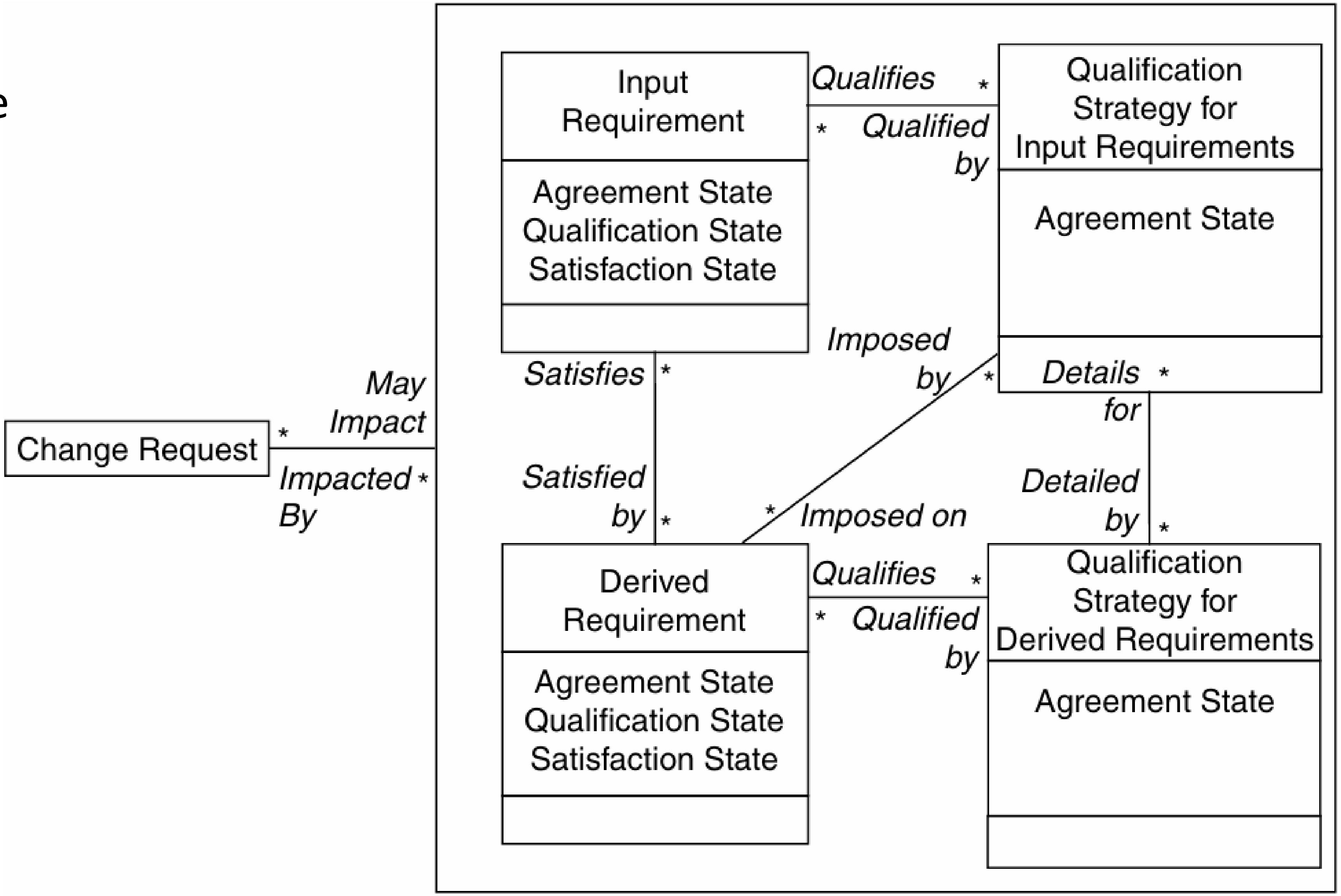
The lines connecting the class symbols show relationships between classes, and these are called **"Associations"** in the UML.



The **asterisk** indicates that zero or more instances of the class can be involved in the association.

Asterisks at both ends indicate that the association can be many to many.

Thus, in figure zero or more input requirements can be satisfied by a derived requirement and an input requirement can be satisfied by zero or more derived requirements.



Input Requirements:

Input requirements specify what information is necessary for the system to perform its tasks accurately.

For example, in a sales system, input requirements might include **customer information, product details,** and **pricin**g.

Derived Requirements:

Derived requirements are additional criteria or data elements that are obtained or calculated from the input requirements.

For example, in a sales system, **derived requirements could be total sales revenue, profit margin, or average order value**.

Qualification Strategy for Input Requirements:

This involves defining how input requirements are identified, validated, and confirmed to ensure their accuracy and completeness.

For example, it includes methods such as user **interviews, surveys, documentation review,** and **prototyping** to gather and validate input requirements.

Additionally, qualification strategies may involve **prioritizing input requirements** based on their importance and impact on the system's functionality..

Qualification Strategy for Derived Requirements:

Similarly, the qualification strategy for derived requirements involves determining how these additional criteria are generated, validated, and confirmed.

For example, It may include **testing algorithms, conducting data analysis,** and **verifying the accuracy of calculations**.

Since derived requirements depend on input requirements, **ensuring the accuracy and reliability of input data** is crucial for validating derived requirements.

Change Request:

- Change requests may arise due to various reasons such as changes in business needs, errors in requirements, or technological advancements.
- When a change request is submitted, it undergoes a review process to assess its impact on the system, including its input and derived requirements.
- The qualification strategies help in evaluating the feasibility and implications of implementing the proposed changes

practical example related to an e-commerce platform

Input Requirements:

Class: Customer Information

Example: This class contains attributes such as customer name, address, email, and phone number. These input requirements are essential for creating a new customer account on the e-commerce platform.

Derived Requirements:

Class: Order Details

Example: Once a customer places an order on the e-commerce platform, the system derives order details such as order ID, order date, total order amount, and shipping address. These derived requirements are calculated based on the input requirements (customer information, product details, and pricing).

Qualification Strategy for Input Requirements:

Class: Input Validation

Example: This class contains methods and rules for validating the input requirements before they are processed by the system.

For instance, the input validation class ensures that the customer's email address follows a valid format, the phone number contains only numeric digits, and the address field is not left blank.

Qualification Strategy for Derived Requirements:

Class: Order Processing

Example: This class contains algorithms and processes for generating and validating derived requirements such as order details and total order amount.

For example, the order processing class calculates the total order amount by summing up the prices of individual items in the customer's shopping cart and applying any applicable discounts or taxes.

Change Request:

Class: Change Management

Example: When there's a change in the pricing structure of products offered on the e-commerce platform, a change request is submitted to update the system's requirements.

The change management class handles the review, approval, and implementation of such change requests, ensuring that they are properly evaluated for their impact on both input and derived requirements.

"satisfies" relationship between input and derived requirement

The "satisfies" relationship between input and derived requirements denotes that the derived requirements are fulfilled or met by the input requirements.

Input Requirement: Customer Account Information

This includes data such as account holder's name, account number, current balance, and transaction history.

Derived Requirement: Account Statement

The account statement is derived from the input requirement of customer account information. It includes details of all transactions (deposits, withdrawals, transfers) made within a specified period.

In this scenario, the derived requirement (account statement) satisfies or is dependent on the input requirement (customer account information). Without the input requirement of customer account information, the system cannot generate an accurate account statement.

"imposed" relationship between derived requirements and qualification strategy for input requirements

- The "imposed" relationship between derived requirements and the qualification strategy for input requirements signifies that the process or methods used to qualify input requirements can influence or dictate the characteristics or properties of the derived requirements.
- In other words, how input requirements are validated or qualified can impose constraints or conditions on the derived requirements.

- **Derived Requirement**: Total Order Amount
    - This is calculated based on the prices of items in a customer's shopping cart, any applicable discounts, and taxes.
- **Qualification Strategy for Input Requirements**: Price Validation
    - This strategy ensures that the prices of items in the shopping cart are accurate, correctly formatted, and within acceptable ranges.

In this scenario, the qualification strategy for input requirements (Price Validation) imposes certain criteria or standards on the input data (item prices).

These criteria directly influence the accuracy and reliability of the derived requirement (Total Order Amount). If the prices of items in the shopping cart are not properly validated or qualified according to the imposed criteria, it can lead to errors or inaccuracies in the calculation of the total order amount.

"detailed by" relationship between qualification strategy for input requirements and qualification strategy for derived requirements

- The "detailed by" relationship between the qualification strategy for input requirements and the qualification strategy for derived requirements signifies that the approach or methods used to qualify input requirements provide detailed guidance or instructions for how to qualify derived requirements.
- In other words, the qualification strategy for input requirements serves as a blueprint or framework for the qualification strategy for derived requirements, providing specific guidelines or procedures for ensuring their accuracy and reliability.

- **Qualification Strategy for Input Requirements**: **Data Validation Framework**
  - This framework outlines the methods and criteria for validating input requirements such as customer data, product information, and pricing details. It specifies procedures for data cleansing, format validation, and range checking.
- **Qualification Strategy for Derived Requirements**: **Calculation Accuracy Guidelines**
  - These guidelines are detailed by the qualification strategy for input requirements and provide specific instructions for validating derived requirements that involve calculations or data processing. For example, in the context of an e-commerce platform, the Calculation Accuracy Guidelines might specify procedures for verifying the accuracy of total order amounts, tax calculations, and discount applications.

In this scenario, the **"detailed by"** relationship indicates that the **Data Validation Framework**, which governs the qualification of input requirements, provides detailed instructions or guidelines that inform the **Calculation Accuracy Guidelines**, which focus on the qualification of derived requirements.

The qualification strategy for input requirements serves as a foundational framework that guides and informs the qualification strategy for derived requirements, ensuring consistency and reliability in the validation process.

A **change request** can apply to any of the other four classes. Enclosing the four classes inside an outer rectangle and making the relationship line touch this outer rectangle indicates this.

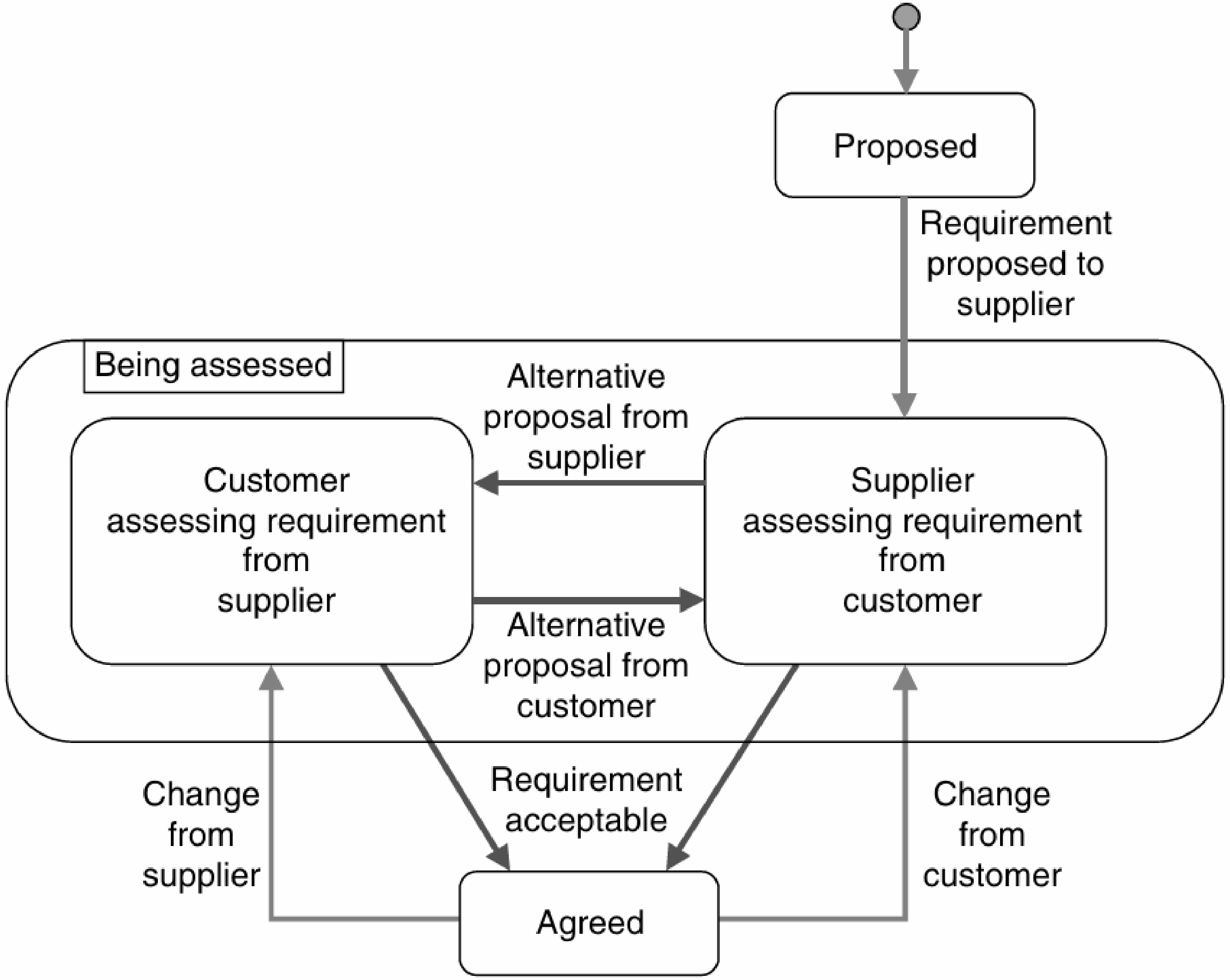
The middle section of the class symbols is used to define attributes that the class will have.

The requirement classes each have the three attributes:

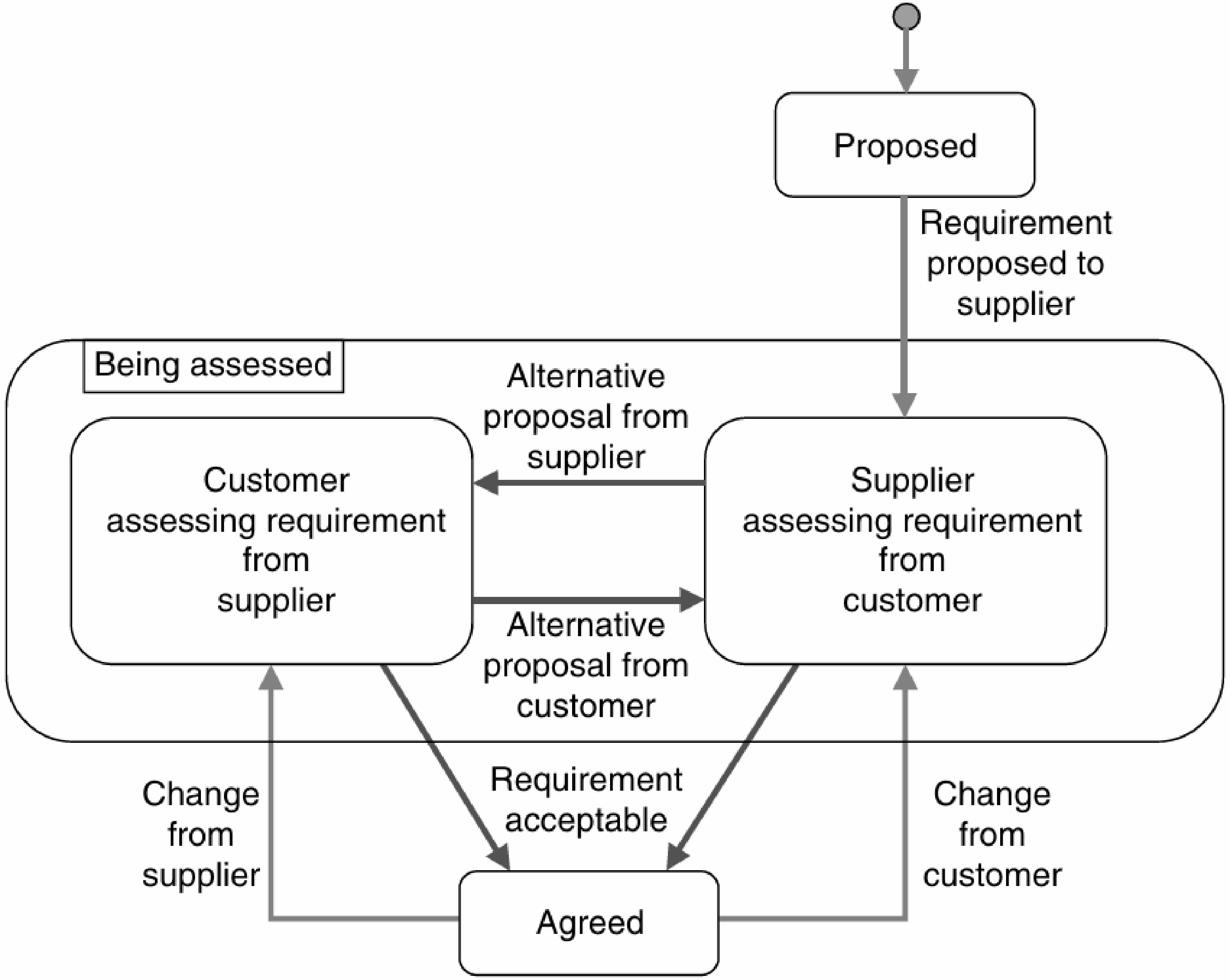
- agreement state
- qualification state
- satisfaction state.

Agreement State

- The rectangle labelled Being assessed is known as a “super-state” because it contains other states within it.
- The lines connecting one state to another indicate transitions that cause the state to change.
- The requirement state starts off in the Proposed state. When the customer is content that the requirement is sufficiently well formulated to be sent to the supplier, it is sent.
- The agreement state then enters the Being assessed super-state. During this state, the customer and supplier negotiate until an agreed requirement emerges


Proposed
Requirement proposed to supplier
Being assessed
Alternative proposal from supplier
Customer assessing requirement from supplier
Supplier assessing requirement from customer
Alternative proposal from customer
Change from supplier
Requirement acceptable
Change from customer
Agreed

- Once in the Agreed state, the requirement will stay there until either the customer or the supplier creates a change request.
- When this happens, the requirement's state re-enters the Being assessed state until a new agreed requirement emerges.
- Within the Being assessed state, the customer and supplier take turns to suggest alternative forms of the requirement until an agreement is reached.
- The agreement state will therefore be in one of the two states shown depending on which party is currently making the assessment


Proposed
Requirement proposed to supplier
Being assessed
Alternative proposal from supplier
Customer assessing requirement from supplier
Supplier assessing requirement from customer
Alternative proposal from customer
Change from supplier
Requirement acceptable
Change from customer
Agreed

In requirements engineering, a **qualification strategy** refers to the approach taken to ensure that the requirements for a system or product are well-defined, complete, and feasible.

It involves methods and techniques for assessing, validating, and prioritizing requirements to ensure that they meet the needs of stakeholders and are suitable for implementation.

Some examples..

Functional Requirements:

- **Example**: In an e-commerce platform, the system shall allow users to add items to their shopping cart.
- **Qualification Strategy**: This requirement can be qualified by ensuring that the system accurately reflects the updated quantity and price of items in the user's shopping cart upon addition.

Performance Requirements:

- **Example**: The system must support a minimum of 1000 concurrent users during peak hours.
- **Qualification Strategy**: This requirement can be qualified by conducting load testing to verify that the system can handle the specified number of concurrent users without significant performance degradation.

Security Requirements:

- **Example**: User passwords must be stored securely using industry-standard encryption algorithms.
- **Qualification Strategy**: This requirement can be qualified by conducting a security audit to ensure that the encryption algorithms used meet industry standards and are resistant to common attacks.

Usability Requirements:

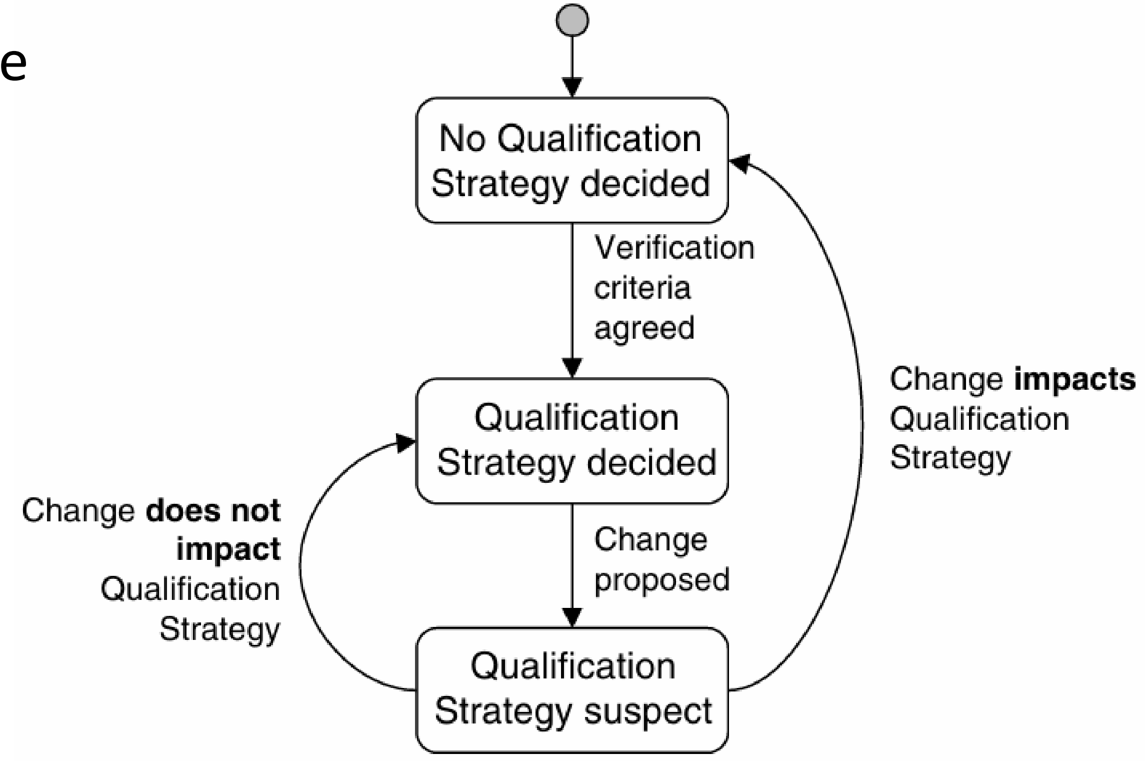
- **Example**: The user interface must be intuitive and easy to navigate.
- **Qualification Strategy**: This requirement can be qualified by conducting usability testing with representative users to evaluate the ease of navigation and overall user experience.

Compatibility Requirements:

- **Example**: The system must be compatible with the latest versions of popular web browsers (Chrome, Firefox, Safari, Edge).
- **Qualification Strategy**: This requirement can be qualified by testing the system's functionality and performance across different web browsers to ensure compatibility.

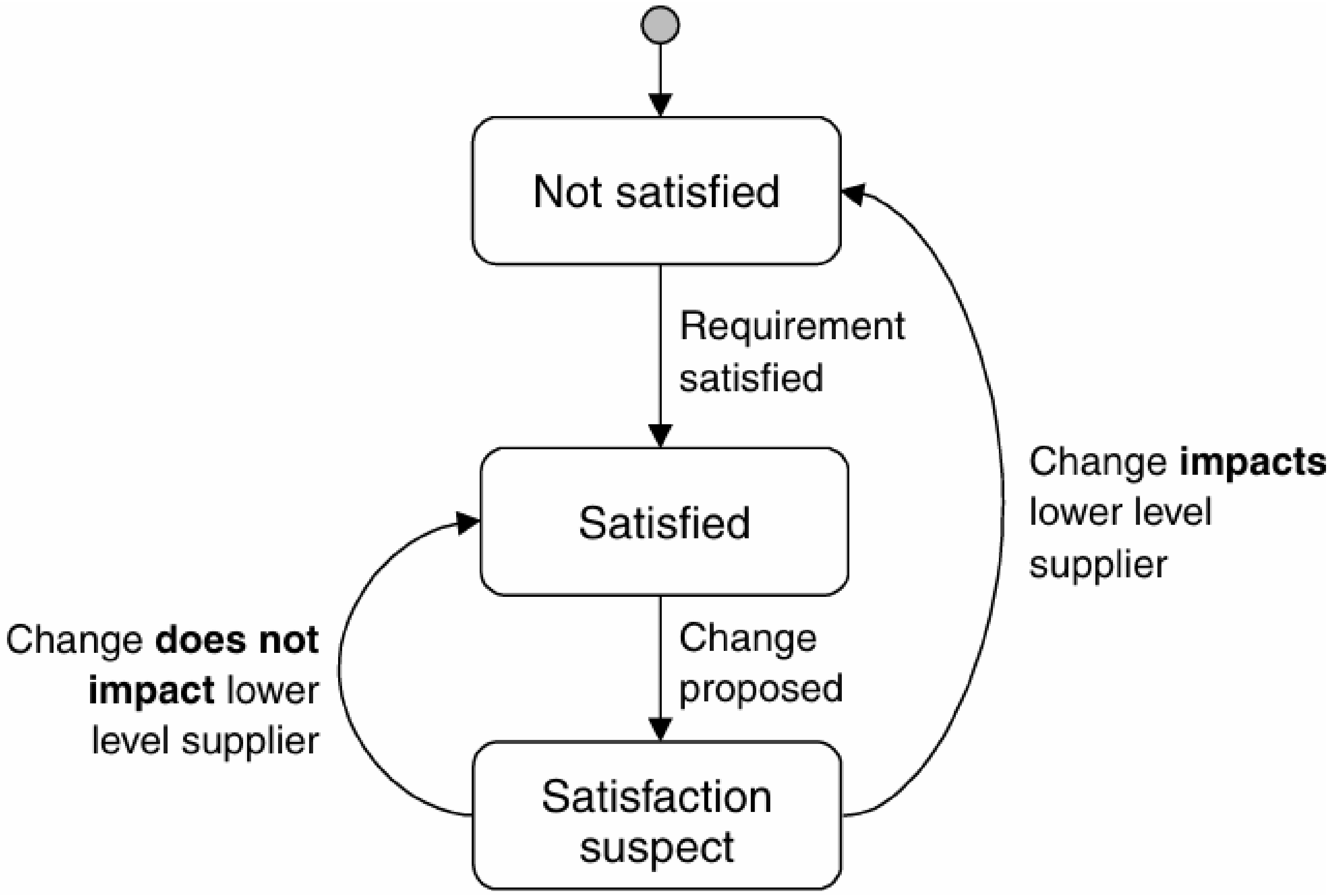
Qualification State

- The initial state is that there is **No qualification strategy decided**. When the qualification strategy has been agreed, the state can proceed to the state **Qualification strategy decided**. This state can then remain until a change request is received.
- The change may be directed either at the requirement itself or at the qualification strategy associated with it.
- When a change is requested, the state becomes **Qualification strategy suspect** until the impact of the change has been assessed. This assessment determines whether the existing qualification strategy can stand, and the state can return to **Qualification strategy decided**, or whether an alternative strategy must be decided, in which case the state becomes **No qualification strategy** decided.

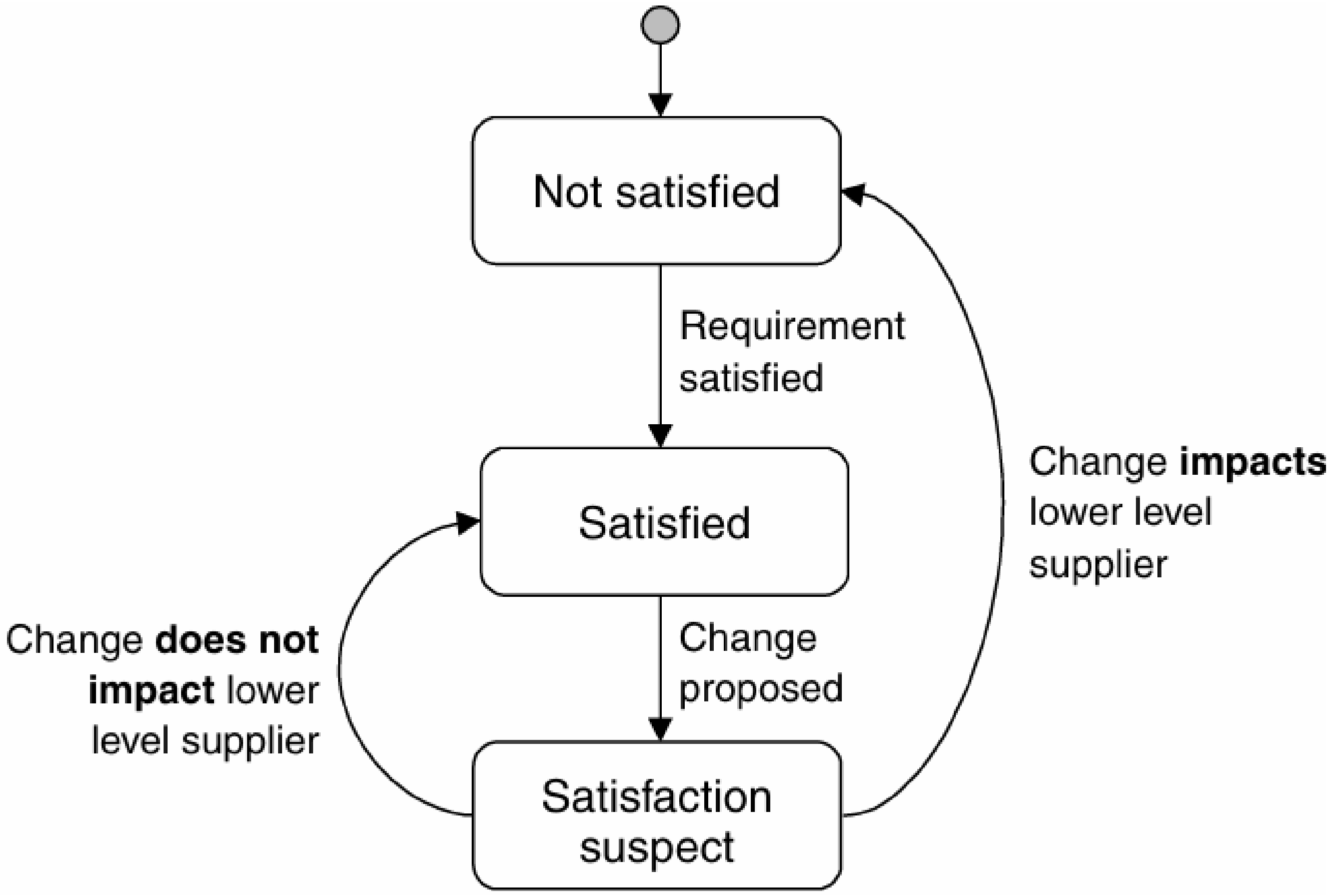

No Qualification Strategy decided
Verification criteria agreed
Qualification Strategy decided
Change proposed
Qualification Strategy suspect
Change does not impact Qualification Strategy
Change impacts Qualification Strategy

Satisfaction State

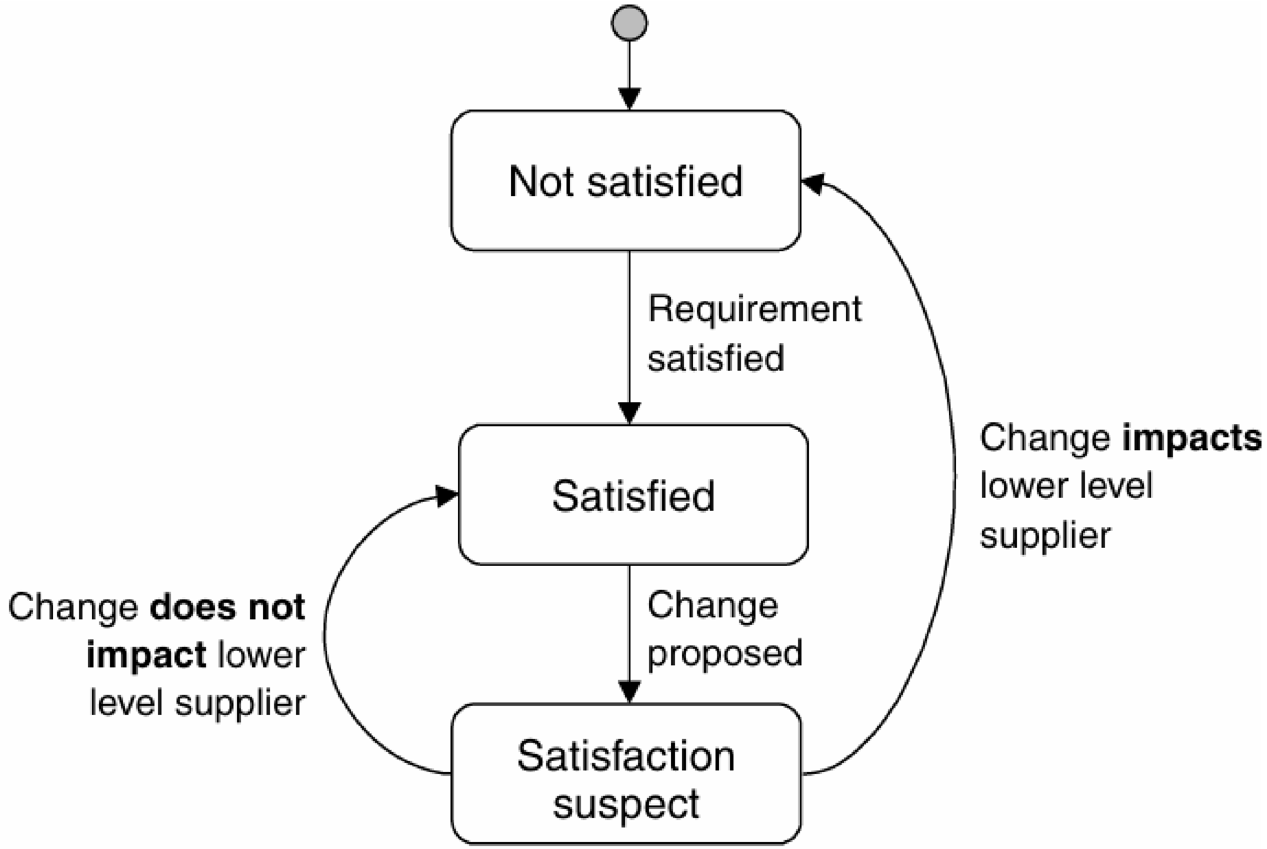
- The starting point is the **Not satisfied state** indicating that no derived requirements have been related to this requirement.


Not satisfied
Requirement satisfied
Satisfied
Change **impacts** lower level supplier
Change **does not impact** lower level supplier
Change proposed
Satisfaction suspect

- When the input requirement has been satisfied by one or more derived requirements (**i.e. higher-level requirement, known as an input requirement, has been met or fulfilled by breaking it down into more detailed, specific requirements, often called derived requirements.),** the lower level supplier agrees the requirement and the higher level (customer) agrees that the derived requirements will, indeed, satisfy the input requirement, the state can be moved to the **Satisfied state.**


Not satisfied
Requirement satisfied
Satisfied
Change impacts lower level supplier
Change does not impact lower level supplier
Change proposed
Satisfaction suspect

- When a change is proposed, the satisfaction state immediately becomes "**Satisfaction suspect**" irrespective of whether the proposed change is directed at the higher or lower level requirements.
- This suspect state is retained until the impact of the proposed change has been assessed and the **satisfaction state** can then become **Not satisfied or Satisfied**.


Not satisfied
Requirement satisfied
Satisfied
Change proposed
Satisfaction suspect
Change **does not impact** lower level supplier
Change **impacts** lower level supplier

Information Model Constraints

- Change requests bind together the agreement, qualification and satisfaction states. Registering a change request immediately changes all three states and requires additional work, first to determine whether there is any impact and second to address the consequences, if any, of the impact.
- Note that the satisfaction state can ripple up and down the requirements that are the subject of the satisfaction relationship.
- This ripple effect establishes the potential extent of any consequential change, that is, the “impact” of the change.
- The agreement state of derived requirements must be consistent with the satisfaction state of input requirements, since an input requirement cannot achieve its satisfied state until the lower level supplier has agreed all of the derived requirements that satisfy it.

Generic Process Details

Agreement Process

The agreement process is always a concurrent activity between a supplier at one level and the customer at the level above

Before any derivation work can commence, it is necessary to assess the input requirements to ascertain whether they form an adequate basis for the development to proceed.

The assessment must answer the questions:

- Is the requirement complete?
- Is the requirement clear?
- Is the requirement implementable?
- Is the qualification plan clear and acceptable?

Potential answers to these questions lead naturally to the following reasons why a requirement may be rejected:

- Missing information – e.g. placeholders such as “TBA”(to be agreed),“TBC” (to be completed) or “TBD”(to be decided) may be used.
- Lack of clarity – ambiguity, contradiction, confusion, etc.
- Impossible to implement – no known solution.
- Unacceptable qualification plan.

Following the review, if a requirement and its qualification plan are acceptable, the status can be set to “Agreed”.

- If the requirement is not acceptable, then an alternative form is sent to the customer and the agreement state becomes “Customer assessing requirement from supplier”. If the customer is content with the alternative wording, then he can set the state to “Agreed”.
- If not, then he or she proposes a further alternative and sends it to the supplier. The agreement state becomes “Supplier assessing requirement from supplier”, and the onus returns to the supplier.
- This process of proposal and counter proposal continues until an agreement is reached. Of course, it is possible that agreement may never be reached and a dispute emerges.
- When either party proposes a change the “Being assessed” super-state is entered with the onus on the party receiving the change.

That’s it